



## **Treball de Fi de Grau**

### **GRAU D'ENGINYERIA INFORMÀTICA**

**Facultat de Matemàtiques i Informàtica  
Universitat de Barcelona**

---

# **IMPLEMENTACIO D'UNA TENDA VIRTUAL AMB TECNOLOGIA DJANGO**

---

**Antoni Rocafort Curia**

Director: Josep Vañó Chic  
Realitzat a: Departament de  
Matemàtiques i Informàtica  
Barcelona, 27 de juny de 2017

# Agraïments

Vull agrair especialment al meu tutor Josep Vañó director del treball final de grau, per la seva dedicació, temps i paciència.

Als professors de les assignatures relacionats amb el treball final de grau i les que no també, per haver transmès els coneixements necessaris per arribar fins aquí i treure aquest projecte endavant.

Gracies a la meva família per creure en el meu recorregut d'estudiant de la UB i ajudar-me en tot en lo que necessiti en aquesta vida. Sense ells no seria possible i no seria la persona que soc avui en dia.

Als meus amics que m'han animat a acabar la carrera, quan tenia l'ànima destruïda i ser capaç de seguir endavant.

# Resum

Aquest projecte es situa en el àmbit del desenvolupament web backend: Implementació de una tenda virtual amb tecnologia django. Concretament es basa en el disseny e implementació de un aplicació web que s'encarrega de gestionar les peticions d'usuaris que li encomanin el usuaris.

La finalitat del projecte es familiaritzar-se amb els web frameworks mes usats del mercat informàtic en aquest cas Django que es un web framework que usa Python. Així a aprendre a fer pàgines web online i crear web servers. S'implementaran mètodes per tenir un major control s'obre el usuaris. Aquests mètodes s'encarregaran principalment de crear, gestionar i borrar, usuaris i productes.

# Abstract

This project is located in the field of web development backend: Implementation of a virtual store with Django technology. Specifically, it is based on the design and implementation of a web application that is responsible for managing user requests that users entrust to it.

The purpose of the project is to relate to the most used web frameworks in the computer market in this case Django, which is a web framework that uses Python. So learn how to make web pages online and create web servers. Methods will be implemented to have greater control open to users. These methods will be primarily responsible for creating, managing and deleting users and products.

# Índex general

## Contenido

Resum.....	3
Contenido .....	5
1.Introducció .....	6
1.1 Objectius.....	7
1.2 Planificació .....	8
1.3 Estat del art .....	9
1.3.1 Web framework .....	9
1.3.2 Que es Django? .....	11
1.4 Enfocament i mètode seguit .....	16
2.Anàlisi .....	17
2.1 Decisions inicials .....	17
2.2 Casos d'us.....	18
3.Planificació.....	32
4.Disseny.....	35
4.1 Arquitectura .....	35
4.2 Prototipat de la web .....	35
4.3 Diagrama de classes .....	42
4.4 Els templates .....	43
5.Implementació .....	46
5.1 Esquema organitzatiu de fitxers.....	46
5.2 El model de dades.....	47
5.2.1 El model relacional .....	47
5.2.2 Les relacions .....	47
5.2.2.1 ManyToManyField .....	47
5.2.2.2 OnetoOneField .....	47
5.2.2.3 ForeignKey .....	48
5.2.3 Classes.....	48
5.2.4 Les consultes o queries .....	49
5.3 La gestió de documents .....	50
5.3.1 Generació, pujada i visualitzar imatges. ....	50
5.4 Interfície .....	51
5.4.1 La aplicació client .....	51
5.4.2 Les views.....	52
5.4.3 Forms .....	53
5.4.4 Les urls .....	54
5.4.5 Carro.....	55
5.4.6 Paypal.....	56
5.4.7 Selecció d'un producte.....	58
5.4.8 Els teus productes.....	59
5.4.9 Les comandes.....	60
6.Conclusions .....	62
6.1 Futures ampliacions .....	62
Glossari.....	64
Bibliografia .....	65

# Index de figures

## Contenido

Figura 1:Funcionament del web framework.....	9
Figura 2: Casos d'usos de usuari .....	19
Figura 3: Casos d'us exclusius del administrador .....	19
Figura 4: Pagina principal de l'aplicatiu web. ....	36
Figura 5 : Pagina de resultats de una cerca .....	37
Figura 6:Visualitzacio de un producte .....	38
Figura 7: Visualització de la pagina de login. ....	39
Figura 8 Visualització de la pagina de registre. ....	40
Figura 9: Visualització de creació de un producte .....	41
Figura 10: Visualització de categories. ....	41
Figura 11: Diagrama de classe .....	42
Figura 12: Pujada de fotos del producte .....	51
Figura 13: Visualització de la aplicació client .....	51
Figura 14: Una view que s'encarrega de crear un producte. ....	53
Figura 15: Uns formularis que s'encarreguen de recollir dades.....	54
Figura 16: Urls amb la seva view corresponent. ....	55
Figura 17: Vista preliminar del carro de compra.....	55
Figura 18: Vista preliminar del procés de pagament en aquest cas les dades de enviament. ....	57
Figura 19: Selecció del mètode de pagament. ....	58
Figura 20: Selecció d'un producte a partir del boto "Buscar" .....	58
Figura 21: Carro de compra amb productes seleccionats.....	59
Figura 22: El anuncis que ha penjat el usuari.....	60
Figura 23: Les comandes o pagaments rebuts a la web.....	60

# 1.Introducció

En aquesta part, serà descrita la idea del projecte que es el funcionament del web backend, que es una part de la informàtica que s'encarrega de gestionar les peticions web que es fa a traves d'un web framework.Principalment la aplicació haurà d'encarregar-se de gestionar una tenda , que els usuaris podran accedir mitjançant el seu explorador web, a mes es podran fer pagaments de manera virtual.

Django es el web framework que s'usarà per aquest projecte, es una tecnologia bastant potent i maniobrable per fer aquesta aplicació.

La tenda virtual està molt de moda avui en dia al mercat informàtic i aquest projecte dona la oportunitat de aprendre moltes coses relacionat amb el mon del marketing i gestió web.

Abans de explicar el TFG , es necessari explicar prèviament uns conceptes que s'usen a lo llarg del procés. Alguns d'ells son trivials però per evitar futures confusions se expliquen algunes idees a continuació.

## 1.1 Objectius

El objectiu d'aquest Treball Final de Grau es crear una aplicació que s'encarregui de gestionar les peticions d'usuari d'una tenda virtual, que fomenti l'ús de la compra-venda de la tenda en si.

Aquest desenvolupament tindrà en compte el volum de dades que volem guardar i gestionar

- Entendre com funcionen els web frameworks.
- Instal·lar les tecnologies necessàries per iniciar el projecte.
- Construir una plantilla HTML.
- Explicar el funcionament Django.
- Disseny de la eina tenda.
- Visualitzar productes de la tenda.
- Registre d'usuaris.
- Gestió de dades.
- Crear un mètode de pagament online.

Per dur a terme aquests objectius es necessiten coneixements bàsics de les assignatures que mencionaré a continuació:

- Factors humans: Utilització del llenguatge HTML per la edició de documents web. CSS per afegir estil a la pàgines.
- Software distribuït: Utilització de web frameworks en aquest cas es Django. Manipulació de arxius tipus Python .py per afegir funcionalitat a les pàgines.

## 1.2 Planificació

El projecte s'ha desenvolupat durant el segon quadrimestre del curs. La implementació de la eina es divideix en tres parts:

Familiaritzar-se amb la tecnologia Django.

- Mirar tutorials i programació Python.
- Entendre el codi.
- Instal·lacions.

Investigació

- Com construir el codi web HTML.
- Investigar el tipus de client que tindrà la tenda.
- la aplicació amb Python.
- Casos d'usos.

Implementació

- Implementar la aplicació amb el mètodes necessaris per la gestió de la tenda.
- Implementació del llenguatge HTML.



## 1.3 Estat del art

### 1.3.1 Web framework

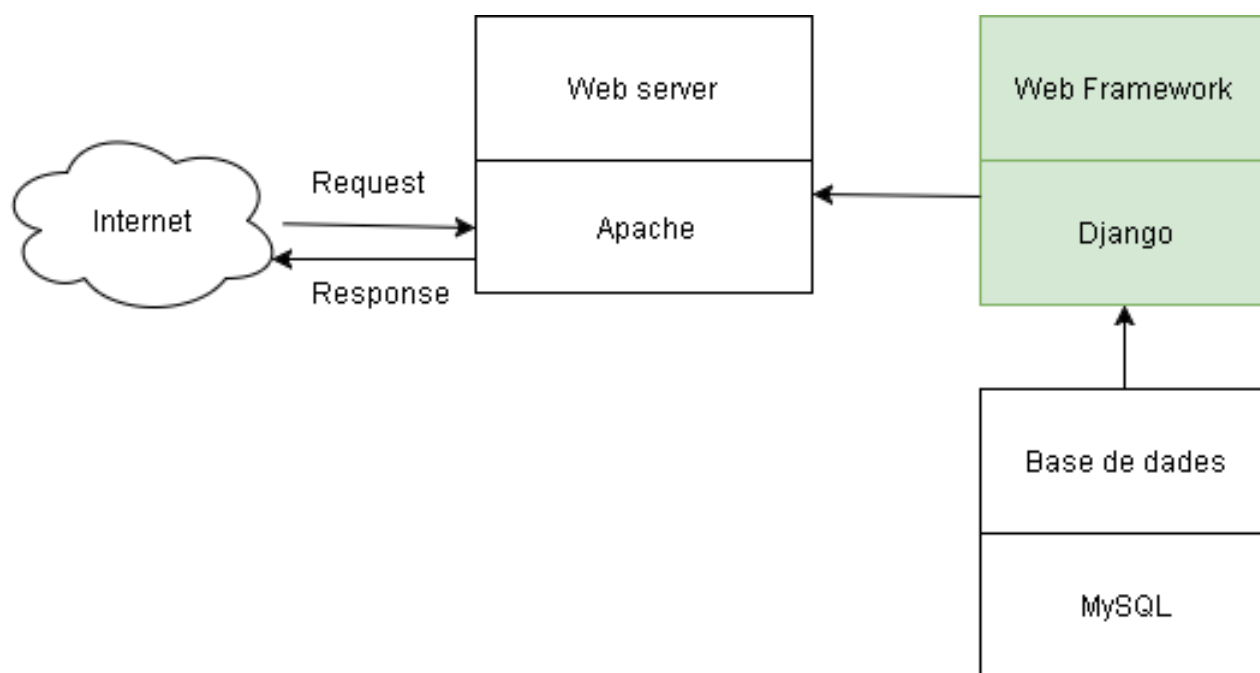


Figura 1:Funcionament del web framework

**Un web framework** es una estructura de programari dissenyada per a donar suport al desenvolupament de llocs web dinàmics, aplicacions web i serveis web. L'estructura del programari intenta reduir l'excés de baixada associats amb activitats comunes usades en desenvolupament web. Per exemple, molts entorns de treball proporcionen llibreries per a l'accés de bases de dades, estructures per a plantilles i gestió de sessions, i amb freqüència faciliten la reutilització de codi.

#### Com triar un web framework

Existeixen molts frameworks web per a casi tots el llenguatges de programació que volem utilitzar. Amb moltes opcions, arriba a ser difícil deduir quin framework proporciona el millor punt de partida per fer la nostra aplicació web.

Alguns dels factors que poden afectar a la nostra decisió son:

**Esforç en l'aprenentatge:** El esforç en l'aprenentatge de un framework web depèn de lo familiaritzat que estiguem amb el llenguatge de programació, la consistència de la seva API, la qualitat de la seva documentació, i el tamany i activitat de la seva comunitat.

**Productivitat:** Productivitat es una solució de quan ràpid pots crear noves característiques una vegada que ens familiaritzem amb el framework, inclosos tan al esforç per escriure com per mantindre el codi. Molts dels factors, que afecten a la productivitat son similars al “Esforç de aprendre”

**Proposit/Origen del framework:** Alguns frameworks web van ser creats inicialment per resoldre certs tipus de problemes, i se mantenen millor en la creació de aplicacions web amb problemàtica similar.

**Dogmàtic versus No dogmàtic:** Un framework dogmàtic es aquell en el que hi ha recomanacions sobre el millor mètode de resoldre un problema en particular. Els frameworks dogmàtics tendeixen a ser mes productius quan estàs tractant de resoldre problemes comuns, perquè et porten a la decisió correcta, encara que son menys flexibles a vegades.

**Amb bateries incloses versus fes-ho tu mateix:** Alguns frameworks webs inclouen eines/biblioteques que resolen per defecte tots els problemes que els seus desenvolupadors puguin pensar, mentrestant els frameworks mes lleugers esperen que els desenvolupadors web triïn i elegir les solucions als seus problemes en biblioteques separades. Els frameworks que inclouen tot son amb mes freqüència mes fàcils de començar amb ells perquè ja tenim tot lo que necessitem i les probabilitats son altes de que estiguin ben integrades i ben documentades. Encara que si un framework mes petit te tot lo que podem necessitar funcionarà en entorns mes estrets i tindran un subconjunt de coses mes petites i mes fàcils de aprendre.

Si el framework potencia o no gaires practiques de desenvolupament: Per exemple, un framework que promou una arquitectura Model-View-Controller per

separar el codi en funcions lògiques resultarà més mantenible que un que no té expectatives en el desenvolupadors. Igualment, el disseny del framework pot tindre un gran impacte en com de fàcil sigui provar i reutilitzar el codi.

**Funció del framework/llenguatge de programació:** Normalment la velocitat no és el factor més gran en la selecció perquè, inclús, amb temps d'execució relativament lents com Python, són més que suficientment bons per a llocs de mida mitja executant-se al hardware moderat. Els beneficis percebuts en velocitat de un altre llenguatge ex. C++ o Javascript, poden veure compensats per els costos d'aprenentatge i manteniment.

**Escalabilitat:** Una vegada que els nostre lloc web tingui un èxit gastarem els beneficis del cache e inclús arribarem als límits del escalat vertical(executar la aplicació en un hardware més potent). En aquest punt podrem necessitar escalar horitzontalment(compartir la carga distribuint el nostre lloc web al llarg de un nombre de servidors webs o base de dades) o escalar “geogràficament” perquè alguns dels nostres clients estan localitzats molt lluny del servidor. El framework web que triem pot marcar una gran diferencia en com de fàcil sigui escalar el nostre lloc web.

**Seguretat web:** Alguns frameworks proporciona millor suport per encarregar-se dels atacs webs més comuns. Django per exemple desinfecta totes les entrades dels usuaris de les plantilles HTML de manera que el possible codi Javascript introduït per el usuari no pugui executar-se. Altres frameworks proporcionen una protecció similar, però no sempre està habilitada per defecte.

### ***1.3.2 Que es Django?***

Django és un framework de desenvolupament web de codi obert, escrit en Python, que respecta el patró Model-Vista-Controlador.

El objectiu de Django és facilitar la creació de llocs webs complexos. Django posa èmfasi en la re-utilització, connectivitat i extensibilitat de components, el ràpid desenvolupament i el principi No et repeteixis. Python és usat en totes les parts del framework, inclús en configuracions, arxius i models de dades.

Els orígens de Django en al administració de pàgines de notícies son evidents al seu disseny, ja que proporciona una sèrie de característiques que faciliten el

rápid desenvolupament de pàgines orientades a continguts. Per exemple , en vegada de requerir que el desenvolupadors escriguin controladors i vistes per les àrees de administració de la pàgina, Django proporciona una aplicació incorporada per administrar els continguts, que pot incloure com part de qualsevol pàgina feta amb Django i que pot administrar varies pàgines fetes amb Django a partir de una mateixa instal·lació; la aplicació administrativa permet la creació, actualització i eliminació de objectes de contingut, portant un registre de totes les accions realitzades a cada un, i proporciona una interfície per administrar usuaris i grups d'usuaris.

Altres característiques de Django son:

Un mapejador objecte-relacional.

Aplicacions que poden instal·lar-se en qualsevol pàgina gestionat per Django.

Una API de base de dades robusta.

Un sistema incorporat de «vistes genèriques» que alhora tindre que escriure la lògica de certes tasques comuns.

Un sistema extensible de plantilles basat en etiquetes, amb herència de plantilles.

Un despatxador de URLS basat en expressions regulars.

Un sistema «middleware» per desenvolupar característiques addicionals.

Suport a la internacionalització

Documentació incorporada accessible a través de la aplicació administrativa.

## **Arquitectura**

Encara que esta fortament inspirat en la filosofia de desenvolupament Model-Vista-Controlador, els seus desenvolupadors declaren públicament que no se senten especialment lligats a observar estrictament ningun paradigma en particular, i en canvi prefereixen fer «lo que sembli correcte».

Gràcies al poder de les capes mediator i foundation, Django permet als desenvolupadors es dediquin a construir els objectes i la lògica de presentació i control per a ells.

## **Presentació**

Aquí es fa la interacció entre usuari i computador. En Django , aquesta tasca la realitza el template engine i el template loader que prenen la informació i la presenten al usuari. El sistema de configuració de Urls es també part de la capa de presentació.

## **Control**

En aquesta capa esta el programa o la lògica de la aplicació en si. En Django son representats per les views i manipuladors. La capa de presentació depèn de aquesta i a la vegada aquesta u fa a la capa de domini.

## **Mediator**

Es el encarregat de manejar la interacció entre el subsistema Entity i Foundation. Aquí es realitza el mapeig objecte-relacional encarregat pel motor de Django.

## **Entity**

El subsistema entity s'encarrega del objectes de negoci. El mapeig objecte-relacional de Django permet escriure objectes de tipo Entity de una forma fàcil i estàndard.

## **Foundation**

La principal tasca del subsistema foundation es encarregar-se a baix nivell el treball de la base de dades. Es proveeix suport a nivell de foundation per a varies bases de dades.

## **Suport a bases de dades.**

Respecte a la base de dades, la recomanada es PostgreSQL, però també son suportades MySQL i SQLite 3. Una vegada creats el data models, Django proporciona una abstracció de base de dades mitjançant de la seva API que permet crear, recuperar, actualitzar i esborrar objectes. També es possible que el usuari executi les seves pròpies consultes SQL directament. En el model de dades de Django, una classe

representa un registre de una taula en la base de dades i les instàncies de aquesta que son les files de la taula.

## **Suport a servidors web**

Com va mencionar al requisits, Django inclou un servidor web per realitzar proves i treballar en la etapa de desenvolupament. En la etapa de producció, es recomana Apache 2 con mod\_python. Encara que Django suporta la especificació WSGI, per lo que pot correr per una sèrie de servidors com FastCGI o SCGI a Apache o altres servidors.

## **Requeriments**

Django necessita Python 2.5 o superior. No es necessiten altres biblioteques de Python per poder obtenir una altra funcionalitat bàsica. En un entorn de desenvolupament no necessitem un web server instal·lat, ja que Django porta el seu servidor per aquest propòsit, amb la restricció de sol permetre un usuari a la vegada.

## **Inconsistències entre la nomenclatura Django i el patró MVC**

Django aparenta implementar el patró MVC, però el controlador es nombrat vista i la vista template.

Primer, hem d'aclarir que al moment de dissenyar Django, no es va buscar apegar a res en particular, sinó en desenvolupar una eina que funcioni lo millor possible.

Si es ben cert que s'assembla molt a la implementació del patró MVC, per a Django la vista descriu «quines» dades seran presentades i no «com» se veuran els mateixos. Aquí es on entren en joc els templates, que descriuen «com les dades son presentades».

Es diu que el «controller» de un MVC clàssic esta representat per el mateix framework. Es a dir, el sistema que envia una request a la vista corresponen, de acord amb la configuració de URL de Django.

## Procés de una petició HTTP

Tenint en compte la arquitectura, veurem com es processa una request.

Quant Django rep una petició Http, lo primer que es fa es crear una instancia de la classe HttpRequest que conte totes les propietats de la petició i diferents mètodes.

Després es realitza la resolució de la URL. Això consisteix en seleccionar la funció de la vista que participarà en la creació de la resposta de la aplicació.

Una vegada que hem resolt quina funció solvent ara la URL especificada, s'invoqui a la funció view amb la instancia **request** de la petició HTTP com a primer paràmetre, el mètode de la vista conte generalment tot el treball lògic, operacions com obtindrà objectes de la base de dades mitjançant models, càlculs, transformacions i finalment la construcció de la representació de la resposta final al usuari.

## Middleware.

Django proveeix tres punts diferents en que permet executar classes middleware, prèviament definides al arxiu de configuració. Una mateixa classe pot executar-se en mes de un punt, aquestes son les opcions:

### Request middleware

S'executa després de crear el objecte HttpRequest, però abans de resoldre la URL, permet modificar el objecte request o tornar una resposta pròpia abans que la resta de la aplicació s'executi.

### View middleware

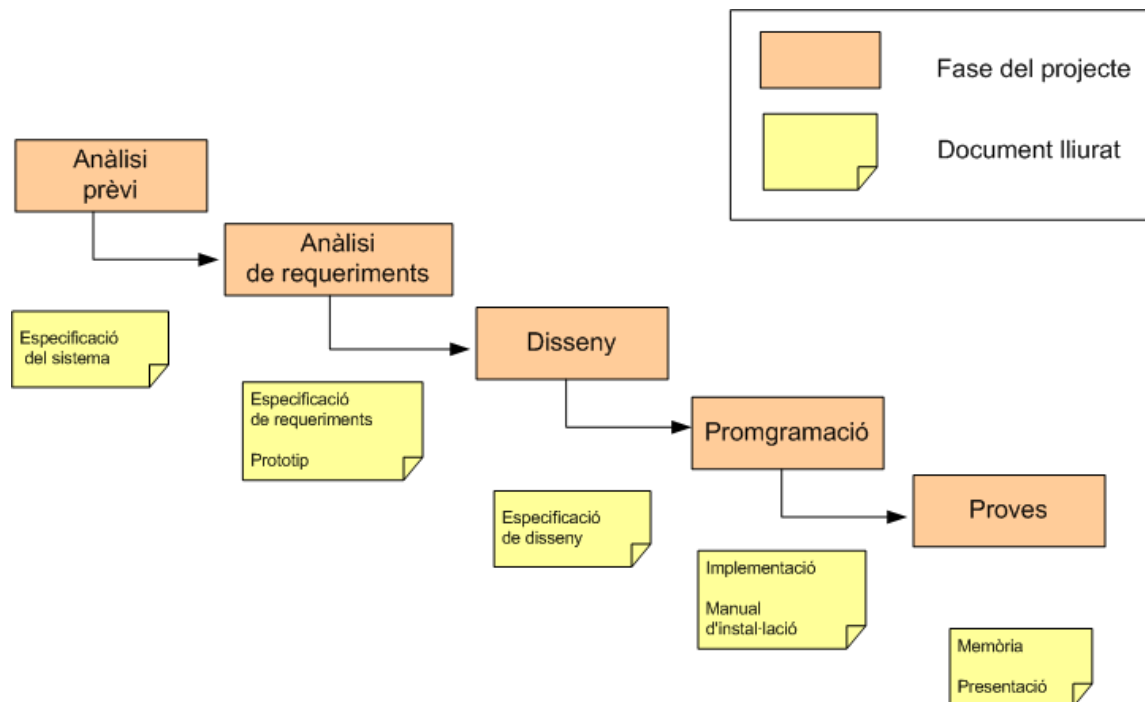
Es executat després de la resolució de la URL, però abans de executar la vista corresponent. Permet executar operacions abans i després de la execució de la vista. La vista pot arribar a no executar-se en absolut.

### Response middleware

S'executa al final, després de que el objecte response hagi sigut creat i abans de entregar al client. Utilitzat per a realitzar les modificacions finals.

## 1.4 Enfocament i mètode seguit

L'enfocament i el mètode seguit per a la realització d'aquest projecte es correspon amb el cicle de vida clàssic d'un projecte (en cascada) i amb prototip. La fase de manteniment no es portarà a terme en aquest treball fi de carrera, donat que està fora de l'àmbit d'aquest treball, en tot cas formaria part d'una fase posterior.





## 2.Anàlisi

### 2.1 Decisions inicials

Abans del inici del desenvolupament de la nostra aplicació web es necessari realitzar un anàlisi per definir temes importants i basics del mateix.

El primer pas a seguir es elegir el tema de la l'aplicació web. En aquest cas, una tenda virtual on disposarà de productes per la compra-venta en si.

En la especificació de requisits es descriuen totes les característiques que ha de complir la web. Per completar la llista de requeriments s'ha investigat sobre les necessitats funcionals que els futurs usuaris de la plataforma necessitarien.

S'han definit dos tipus de usuaris, que se detallen a continuació.

**Usuari:** Consisteix en usuaris que poden haver iniciat sessió o no a la web. Tenen la possibilitat de visualitzar tot el contingut públic del diferents esdeveniments. En cas de iniciar sessió, tenen la possibilitat de editar el seu perfil i proposar nous productes.

**Administrador:** Aquest tipus de usuari es el responsable de la web, ja que pot manipular tot el contingut que se mostra a la plataforma com les de modificar productes.

Després de detallar els diferents actors que existeixen a la plataforma, es llistaran els requeriments funcionals, separats per rols:

Tots el tipus de actors:

- Registrar un compte
- Iniciar sessió
- Editar perfil d'usuari
- Sortir de la sessió
- Consultar tota la informació referida a productes
- Consultar tota la informació referida a categories
- Consultar tota la informació referida a anuncis

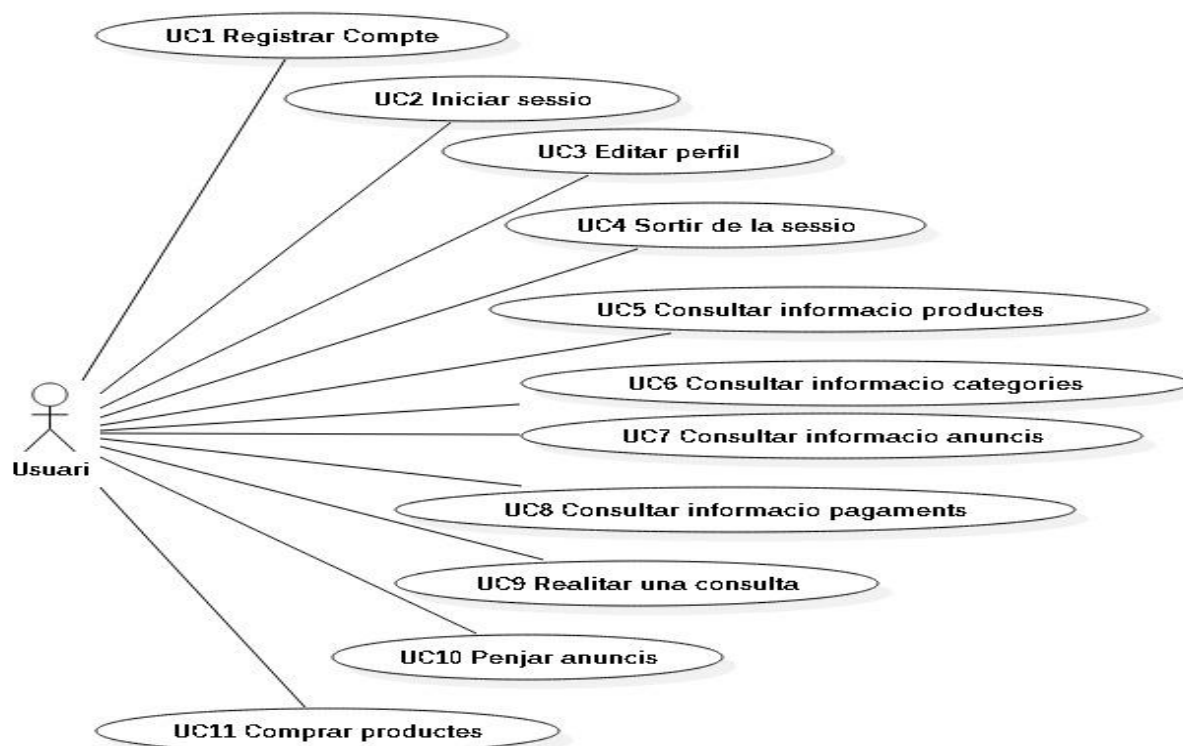
- Consultar tota la informació referida a pagaments
- Realitzar una consulta
- Penjar anuncis
- Comprar productes

Administrador:

- Crear nous usuaris.
- Consultar el llistat de usuaris registrats.
- Editar informació de usuaris registrats, que no siguin administradors.
- Crear, editar, i eliminar tota la informació relacionada amb productes.
- Crear, editar, i eliminar tota la informació relacionada amb categories.
- Crear, editar, i eliminar tota la informació relacionada amb anuncis.
- Crear, editar, i eliminar tota la informació relacionada amb pagaments.
- Crear nous anuncis.

## **2.2 Casos d'us**

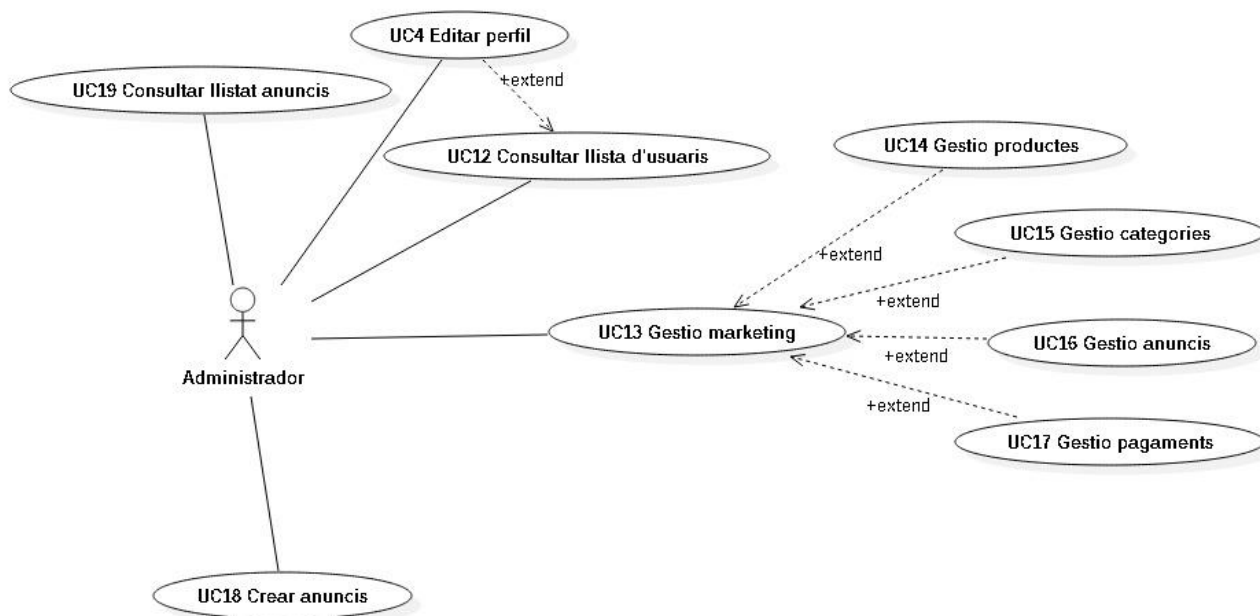
En aquest capítol es detalla les diferents accions que pot realitzar cada tipus de actor , mitjançant dos diagrames d'us i la especificació de cada un d'ells.



**Figura 2: Casos d'usos de usuari**

A la figura 2 es pot observar el llistat de funcionalitats que poden realitzar tant el usuari com el administrador.

A continuació es descriuen el conjunt de casos d'us, mitjançant la seva especificació.



**Figura 3: Casos d'us exclusius del administrador**

En la figura 3 es pot veure el conjunt d'accions que pot realitzar exclusivament el usuari amb el rol de administrador.

<b>UC01</b>	<b>Registre del compte</b>
<b>Descripció</b>	Permet registrar-se a la aplicació.
<b>Actors</b>	Usuari i administrador
<b>Pre-condicions</b>	<p>Els actors han de accedir a la pagina principal.</p> <p>No tenen que haver iniciat sessió.</p> <p>El nom de usuari i el correu electrònic que introdueixi el usuari no ha de estar registrat.</p>
<b>Flux basic</b>	
	<ol style="list-style-type: none"> <li>1. El actor selecciona el boto Sign up situat a la part superior de la pagina.</li> <li>2. El sistema mostra el formulari de registre de usuaris.</li> <li>3. El actor introdueix tots el camps obligatoris del formulari i selecciona el boto Register.</li> <li>4. El sistema envia un correu electrònic, al compte indicat en el formulari anterior, amb un enllaç per activar el compte.</li> <li>5. El sistema envia un correu electrònic, al compte indicat en el formulari anterior, amb un enllaç per activar el compte.</li> <li>6. El actor selecciona el enllaç del correu electrònic.</li> <li>7. El sistema mostra un missatge indicant que s'ha activat el compte amb èxit.</li> </ol>
<b>Post-condicions</b>	Es mostra un missatge per indicar que el actor ja pot iniciar sessió, amb les dades introduïdes.

<b>UC02</b>	<b>Iniciar sessió</b>
<b>Descripció</b>	Permet que els actors iniciïn sessió a la plataforma.
<b>Actors</b>	Usuari i administrador
<b>Pre-condicions</b>	<p>Els actors han de accedir a la pagina principal.</p> <p>No tenen que haver iniciat sessió.</p> <p>Tenen que haver utilitzat UC01: Registre de compte</p>
<b>Flux basic</b>	
	<ol style="list-style-type: none"> <li>1. El actor selecciona el boto Sign up situat a la part superior de la pagina.</li> <li>2. El sistema mostra el formulari de dos camps per iniciar sessió.</li> <li>3. El actor introdueix el nom de usuari i la contrasenya, i selecciona el boto Sign in</li> <li>4. El sistema mostra la pagina principal.</li> </ol>
<b>Post-condicions</b>	Es mostra la pagina principal, però amb la diferencia que ara, en la part superior de la pagina, es pot observar un menu de usuari.

<b>Flux alternatiu.</b>
<p>4.a El actor ha introduït dades incorrectes.</p> <p>El sistema mostra un missatge de error.</p> <p>El client torna al pas 3.</p>
<p>4.b El actor no recorda la contrasenya.</p> <p>El client selecciona la opció Forgot password?</p>
<b>Tabla 2 : Cas d'us UC02</b>

<b>UC03</b>	<b>Sortir de la sessió</b>
<b>Descripció</b>	Permet als actors sortir de la sessió iniciada.
<b>Actors</b>	Usuari i administrador
<b>Pre-condicions</b>	Els actors han de haver realitzat UC02: Iniciar sessió.
<b>Flux basic</b>	
<ol style="list-style-type: none"> <li>1. El actor selecciona el menu de usuari amb el seu nom situat a la part superior de la pagina.</li> <li>2. El sistema mostra un desplegable amb diferents opcions.</li> <li>3. El actor selecciona la opció Logout</li> </ol>	
<b>Post-condicions</b>	Es mostra la pagina principal.
<b>Tabla 3: Cas d'us UC03</b>	

<b>UC04</b>	<b>Editar perfil</b>
<b>Descripció</b>	Permet als actors editar la seva informació.
<b>Actors</b>	Usuari i administrador
<b>Pre-condicions</b>	Els actors han de haver utilitzat UC02: Iniciar sessió.
<b>Flux basic</b>	
<ol style="list-style-type: none"> <li>1. El actor selecciona el menú de usuari amb el seu nom situat a la part superior de la pagina.</li> <li>2. El sistema mostra un desplegable amb diferents opcions.</li> <li>3. El actor selecciona la opció Edit perfil-le.</li> <li>4. El sistema mostra un formulari replet amb la informació introduïda per el actor en el registre.</li> <li>5. El actor modifica les opcions que desitgi i selecciona el boto Save.</li> </ol>	
<b>Post-condicions</b>	Es mostra la pagina principal, amb les dades guardades.

**Tabla 4: Cas d'us UC04**

<b>UC05</b>	<b>Consultar informació de productes.</b>
<b>Descripció</b>	<b>Permet als actors observar tota la informació publica referida a productes.</b>
<b>Actors</b>	<b>Usuari i administrador</b>
<b>Pre-condicions</b>	<b>Els actors han de accedir a la pagina principal.</b>
<b>Flux basic</b>	
	<ol style="list-style-type: none"> <li>1. El actor selecciona el menú nombrat Productes situat a la part superior de la pagina.</li> <li>2. El sistema mostra la pantalla amb la llista de productes.</li> <li>3. El actor selecciona un producte.</li> <li>4. El sistema mostra una pagina amb la descripció del producte i el boto comprar.</li> <li>5. El actor pot accedir a la diferent informació del producte i si vol pot comprar el producte.</li> </ol>
<b>Post-condicions en concret.</b>	<b>Es mostra una pagina que detalla informació de un producte</b>
<b>Taula 5:Cas d'us UC05</b>	

<b>UC06</b>	<b>Consultar informació de categories.</b>
<b>Descripció</b>	<b>Permet als actors observar tota la informació publica referida a categories.</b>
<b>Actors</b>	<b>Usuari i administrador</b>
<b>Pre-condicions</b>	<b>Els actors han de accedir a la pagina principal.</b>
<b>Flux basic</b>	
	<ol style="list-style-type: none"> <li>1. El actor selecciona una categoria que mostra un desplegable .</li> <li>2. El sistema mostra la pantalla una llista de productes, depenen de la categoria que hagi elegit l'usuari.</li> <li>3. El actor selecciona un producte.</li> <li>4. El sistema mostra una pagina amb la descripció del producte i el boto comprar.</li> <li>5. El actor pot accedir a la diferent informació del producte i si vol pot comprar el producte.</li> </ol>
<b>Post-condicions</b>	<b>Es mostra una pagina que detalla informació de una categoria en concret.</b>
<b>Taula 6:Cas d'us UC06</b>	

<b>UC07</b>	<b>Consultar informació de anuncis.</b>
<b>Descripció</b>	Permet als actors observar tota la informació publica referida a anuncis.
<b>Actors</b>	Usuari i administrador
<b>Pre-condicions</b>	Els actors han de accedir a la pagina principal.
<b>Flux basic</b> <ol style="list-style-type: none"> <li>1. El actor selecciona el menú nombrat Anuncis situat a la part superior de la pagina.</li> <li>2. El sistema mostra la pantalla amb la llista de anuncis.</li> <li>3. El actor selecciona un anunci.</li> <li>4. El sistema mostra una pagina amb la descripció del anunci.</li> <li>5. El actor pot editar la informació del anunci.</li> </ol>	
<b>Post-condicions en concret.</b>	Es mostra una pagina que detalla informació de un anunci
<b>Taula 7: Cas d'us UC07</b>	

<b>UC08</b>	<b>Consultar informació de pagaments.</b>
<b>Descripció</b>	Permet als actors observar tota la informació publica referida a pagaments.
<b>Actors</b>	Usuari i administrador
<b>Pre-condicions</b>	Els actors han de accedir a la pagina principal.
<b>Flux basic</b> <ol style="list-style-type: none"> <li>1. El actor selecciona el menú nombrat pagaments situat a la part superior de la pagina.</li> <li>2. El sistema mostra la pantalla amb la llista de pagaments.</li> <li>3. El actor selecciona un pagament.</li> <li>4. El sistema mostra una pagina amb la descripció del pagament.</li> <li>5. El actor pot cancel·lar un pagament realitzat.</li> </ol>	
<b>Post-condicions</b>	Es mostra una pagina que detalla informació de un pagament en concret.
<b>Taula 8: Cas d'us UC08</b>	

<b>UC09</b>	<b>Realitzar una consulta.</b>
-------------	--------------------------------

<b>Descripció</b>	<b>Permet als actors realitzar una consulta mitjançant una caixa de entrada de text.</b>
<b>Actors</b>	<b>Usuari i administrador</b>
<b>Pre-condicions</b>	<b>Els actors han de accedir a la pagina principal.</b>
<b>Flux basic</b>	
	<ol style="list-style-type: none"> <li>1. El actor introdueix algun caràcter a buscar a la caixa de entrada del text situat a la part superior de la pagina, i selecciona la tecla Enter.</li> <li>2. El sistema mostra una llista de productes amb el resultat de la consulta.</li> <li>3. El actor selecciona un resultat.</li> <li>4. El sistema mostra una pagina amb la descripció del producte i el boto comprar.</li> </ol>
<b>Taula 9:Cas d'us UC09</b>	

<b>UC010</b>	<b>Penjar anunci.</b>
<b>Descripció</b>	<b>Permet als actors penjar anuncis de productes a la pagina.</b>
<b>Actors</b>	<b>Usuari i administrador</b>
<b>Pre-condicions</b>	<b>Els actors han de accedir a la pagina principal.</b> <b>Els actors han hauran realitzat :UC2 Iniciar sessió.</b>
<b>Flux basic</b>	
	<ol style="list-style-type: none"> <li>1. El actor selecciona el menú de usuari amb el seu nom situat a la part superior de la pagina.</li> <li>2. El sistema mostra un desplegable amb diferents opcions.</li> <li>3. El actor selecciona Create advert.</li> <li>4. El sistema mostra un formulari.</li> <li>5. El actor introdueix el tipo de producte, categoria, descripció i preu, i selecciona el boto Send.</li> <li>6. El sistema mostra la pagina principal.</li> </ol>
<b>Taula 10:Cas d'us UC010</b>	

<b>UC011</b>	<b>Comprar productes</b>
--------------	--------------------------



<b>Descripció</b>	<b>Permet als actors realitzar una compra mitjançant un boto.</b>
<b>Actors</b>	<b>Usuari i administrador</b>
<b>Pre-condicions</b>	<b>Els actors han de accedir a la pagina principal.</b>
<b>Flux basic</b>	<ol style="list-style-type: none"> <li>1. El actor clica sobre el boto comprar del producte en concret .</li> <li>2. El sistema mostra el sistema de pagament.</li> <li>3. El actor selecciona un pagament.</li> <li>4. El sistema mostra la pagina principal amb un missatge que s'ha completat el pagament.</li> </ol>
<b>Taula 11:Cas d'us UC011</b>	

<b>UC012</b>	<b>Consultar llista de usuaris</b>
<b>Descripció la plataforma.</b>	<b>Permet als actors consultar la llista de usuaris registrats a</b>
<b>Actors</b>	<b>Administrador</b>
<b>Pre-condicions</b>	<b>Els actors han de accedir a la pagina principal.</b> <b>Els actors hauran realitzat UC02: Iniciar sessió.</b>
<b>Flux basic</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona el menú de usuari amb el seu nom situat a la part superior de la pagina.</li> <li>2. El sistema mostra un desplegable amb diferents opcions.</li> <li>3. El actor selecciona la opció User zone dintre de Admin site.</li> <li>4. El sistema mostra un llistat de usuaris.</li> </ol>
<b>Post- condicions</b>	<b>Es mostra una pagina amb la llista de usuaris.</b>
<b>Flux alternatiu</b>	<b>5.a El actor vol editar el perfil de un usuari no administrador UC04</b>
<b>Taula 12:Cas d'us UC012</b>	

<b>UC013</b>	<b>Gestió marketing</b>
<b>Descripció</b>	Permet als actors crear, editar i eliminar productes, categories, anuncis i pagaments.
<b>Actors</b>	Administrador
<b>Pre-condicions</b>	Els actors han de accedir a la pagina principal. Els actors hauran realitzat UC02: Iniciar sessió.
<b>Flux basic</b> <ol style="list-style-type: none"> <li>1. El actor selecciona el menú de usuari amb el seu nom situat a la part superior de la pagina.</li> <li>2. El sistema mostra un desplegable amb diferents opcions.</li> <li>3. El actor selecciona la opció Marketing zone dintre de Admin site.</li> <li>4. El sistema mostra un llistat de productes creats.</li> </ol>	
<b>Post- condicions</b>	El sistema realitza alguna gestió sobre un producte.
<b>Flux alternatiu</b> <p><b>3.a El actor vol crear un nou producte.</b>  El actor selecciona el boto Afegir, per a crear un nou producte.  El sistema mostra un formulari.  El actor introdueix títol, categoria , descripció i preu.  El sistema torna al pas 3.</p> <p><b>3.b El actor vol eliminar un producte</b>  El actor selecciona el boto Eliminar, per eliminar el producte.  El sistema mostra una finestra, demanat confirmació.  El actor selecciona Delete.  El sistema torna al pas 3.</p> <p><b>3.c El actor vol editar un producte.</b>  El actor selecciona el nom del producte.  El sistema mostra una finestra, on es pot editar tot el contingut del producte seleccionat.  El actor pot modificar el títol i la descripció del producte i a mes. <ul style="list-style-type: none"> <li>• Gestionar els usuaris relacionats amb un producte.</li> <li>• Gestionar les categories de un producte.</li> <li>• Gestionar els pagaments de un producte.</li> <li>• Gestionar els anuncis de un producte.</li> </ul> El actor selecciona el boto Guardar  El sistema torna al pas 3.</p>	
<b>Taula 13:Cas d'us UC013</b>	

<b>UC014</b>	<b>Gestió productes</b>
<b>Descripció</b>	<b>Permet als actors crear, editar i eliminar productes.</b>
<b>Actors</b>	<b>Administrador</b>
<b>Pre-condicions</b>	<b>Els actors han de accedir a la pagina principal. Els actors hauran realitzat UC02: Iniciar sessió.</b>
<p><b>Flux basic</b></p> <ol style="list-style-type: none"> <li>1. El actor selecciona el menú de usuari amb el seu nom situat a la part superior de la pagina.</li> <li>2. El sistema mostra un desplegable amb diferents opcions.</li> <li>3. El actor selecciona la opció Products zone dintre de Admin site.</li> <li>4. El sistema mostra un llistat de productes creats.</li> </ol>	
<b>Post- condicions</b>	<b>El sistema realitza alguna gestió sobre un producte.</b>
<p><b>Flux alternatiu</b></p> <p><b>3.a El actor vol crear un nou producte.</b>  El actor selecciona el boto Afegir, per a crear un nou producte.  El sistema mostra un formulari.  El actor introdueix títol, categoria , descripció i preu.  El sistema torna al pas 3.</p>	
<p><b>3.b El actor vol eliminar un producte</b>  El actor selecciona el boto Eliminar, per eliminar el producte.  El sistema mostra una finestra, demanat confirmació.  El actor selecciona Delete.  El sistema torna al pas 3.</p>	
<p><b>3.c El actor vol editar un producte.</b>  El actor selecciona el nom del producte.  El sistema mostra una finestra, on es pot editar tot el contingut del producte seleccionat.  El actor pot modificar el títol i la descripció del producte i a mes. <ul style="list-style-type: none"> <li>▪ Gestionar els usuaris relacionats amb un producte.</li> <li>▪ Gestionar les categories de un producte.</li> <li>▪ Gestionar els pagaments de un producte.</li> <li>▪ Gestionar els anuncis de un producte.</li> </ul> El actor selecciona el boto Guardar  El sistema torna al pas 3.</p>	

**Taula 14: Cas d'us UC014**

<b>UC015</b>	<b>Gestió categories</b>
<b>Descripció</b>	<b>Permet als actors crear, editar i eliminar categories.</b>
<b>Actors</b>	<b>Administrador</b>
<b>Pre-condicions</b>	<b>Els actors han de accedir a la pagina principal. Els actors hauran realitzat UC02: Iniciar sessió.</b>
<b>Flux basic</b>	
	<ol style="list-style-type: none"> <li>1. El actor selecciona el menú de usuari amb el seu nom situat a la part superior de la pagina.</li> <li>2. El sistema mostra un desplegable amb diferents opcions.</li> <li>3. El actor selecciona la opció Categories zone dintre de Admin site.</li> <li>4. El sistema mostra un llistat de categories creades.</li> </ol>
<b>Post- condicions</b>	<b>El sistema realitza alguna gestió sobre una categoria.</b>
<b>Flux alternatiu</b>	
<b>3.a El actor vol crear un nou producte.</b> El actor selecciona el boto Afegir, per a crear una nova categoria. El sistema mostra un formulari El actor introdueix nom de la categoria. El sistema torna al pas 3.	
<b>3.b El actor vol eliminar un producte</b> El actor selecciona el boto Eliminar, per eliminar la categoria. El sistema mostra una finestra, demanat confirmació. El actor selecciona Delete. El sistema torna al pas 3.	
<b>3.c El actor vol editar una categoria.</b> El actor selecciona el nom de la categoria. El sistema mostra una finestra, on es pot editar tot el contingut de la categoria seleccionada. El actor pot modificar el nom. El actor selecciona el boto Guardar El sistema torna al pas 3.	
<b>Taula 15: Cas d'us UC015</b>	

<b>UC016</b>	<b>Gestió anuncis</b>
<b>Descripció</b>	Permet als actors crear, editar i eliminar anuncis.
<b>Actors</b>	Administrador
<b>Pre-condicions</b>	Els actors han de accedir a la pagina principal. Els actors hauran realitzat UC02: Iniciar sessió.
<b>Flux basic</b> <ol style="list-style-type: none"> <li>1. El actor selecciona el menú de usuari amb el seu nom situat a la part superior de la pagina.</li> <li>2. El sistema mostra un desplegable amb diferents opcions.</li> <li>3. El actor selecciona la opció Adds zone dintre de Admin site.</li> <li>4. El sistema mostra un llistat de anuncis creats.</li> </ol>	
<b>Post- condicions</b>	El sistema realitza alguna gestió sobre un anunci.
<b>Flux alternatiu</b> <p><b>3.a El actor vol crear un nou producte.</b>  El actor selecciona el boto Afegir, per a crear un nou anunci.  El sistema mostra un formulari.  El actor introdueix títol, categoria , descripció i preu.  El sistema torna al pas 3.</p> <p><b>3.b El actor vol eliminar un anunci</b>  El actor selecciona el boto Eliminar, per eliminar el anunci.  El sistema mostra una finestra, demanat confirmació.  El actor selecciona Delete.  El sistema torna al pas 3.</p> <p><b>3.c El actor vol editar un anunci.</b>  El actor selecciona el nom del anuncis.  El sistema mostra una finestra, on es pot editar tot el contingut del anunci seleccionat.  El actor pot modificar el títol i la descripció del anunci.  El actor selecciona el boto Guardar.  El sistema torna al pas 3.</p>	
<b>Taula 16:Cas d'us UC016</b>	

<b>UC017</b>	<b>Gestió pagaments</b>
<b>Descripció</b>	Permet als actors crear, editar i eliminar pagaments.
<b>Actors</b>	Administrador
<b>Pre-condicions</b>	Els actors han de accedir a la pagina principal. Els actors hauran realitzat UC02: Iniciar sessió.
<p><b>Flux basic</b></p> <ol style="list-style-type: none"> <li>1. El actor selecciona el menú de usuari amb el seu nom situat a la part superior de la pagina.</li> <li>2. El sistema mostra un desplegable amb diferents opcions.</li> <li>3. El actor selecciona la opció Payments zone dintre de Admin site.</li> <li>4. El sistema mostra un llistat de pagaments creats.</li> </ol>	
<b>Post- condicions</b>	El sistema realitza alguna gestió sobre un pagament.
<p><b>Flux alternatiu</b></p> <p><b>3.a El actor vol crear un nou pagament.</b>  El actor selecciona el boto Afegir, per a crear un nou pagament.  El sistema mostra un formulari.  El actor introdueix títol, data, categoria , descripció i preu.  El sistema torna al pas 3.</p> <p><b>3.b El actor vol eliminar un pagament.</b>  El actor selecciona el boto Eliminar, per eliminar el pagament.  El sistema mostra una finestra, demanat confirmació.  El actor selecciona Delete.  El sistema torna al pas 3.</p> <p><b>3.c El actor vol editar un pagament.</b>  El actor selecciona el nom del pagament.  El sistema mostra una finestra, on es pot editar tot el contingut del pagament seleccionat.  El actor pot modificar el títol, data i la descripció del pagament.  El actor selecciona el boto Guardar  El sistema torna al pas 3.</p>	
Taula 17:Cas d'us UC017	

<b>UC018</b>	<b>Crear anuncis</b>
--------------	----------------------

<b>Descripció</b>	<b>Permet als actors crear anuncis</b>
<b>Actors</b>	<b>Administrador</b>
<b>Pre-condicions</b>	<b>Els actors han de accedir a la pagina principal. Els actors hauran realitzat UC02: Iniciar sessió.</b>
<b>Flux basic</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona el menú de usuari amb el seu nom situat a la part superior de la pagina.</li> <li>2. El sistema mostra un desplegable amb diferents opcions.</li> <li>3. El actor selecciona la opció Adds zone dintre de Admin site.</li> <li>4. El sistema mostra un llistat de anuncis creats.</li> </ol>
<b>Post- condicions</b>	<b>El sistema realitza alguna gestió sobre un pagament.</b>
<b>Flux alternatiu</b>	<p><b>3.a El actor vol crear un nou anuncis</b></p> <p>El actor selecciona el boto Afegir, per a crear un nou pagament.</p> <p>El sistema mostra un formulari.</p> <p>El actor introdueix títol, data, categoria , descripció i preu.</p> <p>El sistema torna al pas 3.</p>

**Taula 18:Cas d'us UC018**

### 3. Planificació

**GanttProject [gant.gan]**

Project Edit View Tasks Resources Help

Gantt Resources Chart

**GANTT project**

Name	Begin date	End date	Duration
• Rummi tenda	2/5/18	6/1/18	85
• Presentació	2/5/18	2/7/18	3
• Introducció	2/5/18	2/5/18	1
• Motivacions	2/6/18	2/7/18	2
• Lliurament Presentació	2/8/18	2/8/18	0
• Planificació	2/8/18	2/12/18	3
• Lliurament Planificació	2/13/18	2/13/18	0
• Pla de Projecte	2/13/18	2/20/18	6
• Redactar Pla Projecte	2/13/18	2/20/18	6
• Lliurament Pla de Projecte	2/21/18	2/21/18	0
• Anàlisi i disseny	2/21/18	3/28/18	26
• Elaborar anàlisi	2/21/18	3/9/18	13
• Elaborar disseny	3/12/18	3/28/18	13
• Lliurament anàlisi i disseny	3/29/18	3/29/18	0
• Implementació	3/29/18	5/22/18	39
• Productes	3/29/18	4/19/18	16
• Anuncis	4/20/18	5/2/18	9
• Clients	5/3/18	5/11/18	7
• Proves integració	5/14/18	5/22/18	7
• Lliurament Implementació	5/23/18	5/23/18	0
• Memòria i presentació	5/23/18	6/1/18	8
• Memòria	5/23/18	5/28/18	4
• Presentació	5/29/18	6/1/18	4
• Presentació usuaris	5/23/18	5/30/18	6
• Formació	5/23/18	5/30/18	6
• Final Projecte	6/4/18	6/4/18	0



Data Lliurament	Document	Descripció del document
<b>25 de febrer</b>	Pla de projecte	És el pla de treball en el qual es detalla la descripció, els requeriments previs i la planificació temporal del projecte.
<b>28 de Març</b>	Anàlisi i disseny	L'anàlisi i el disseny previs a la implementació. Són parts fonamentals del projecte i cal tractar-les en profunditat abans de començar la implementació.
<b>15 de Maig</b>	Implementació	L'aplicació final completa i funcional.
<b>27 de Juny</b>	Memòria i presentació	A la memòria del projecte es sintetitzarà el treball realitzat durant el projecte, es mostrarà l'assoliment dels objectius proposats i s'indicaran les conclusions obtingudes. La presentació pretén abastar les parts del projecte més interessants i ser un element dinàmic i atractiu en el qual es pugui recolzar l'exposició final.



## 4.Disseny

En aquest capítol es detallen les diferents fases del disseny i desenvolupament de l'aplicació, seguint un ordre cronològic.

### 4.1 Arquitectura

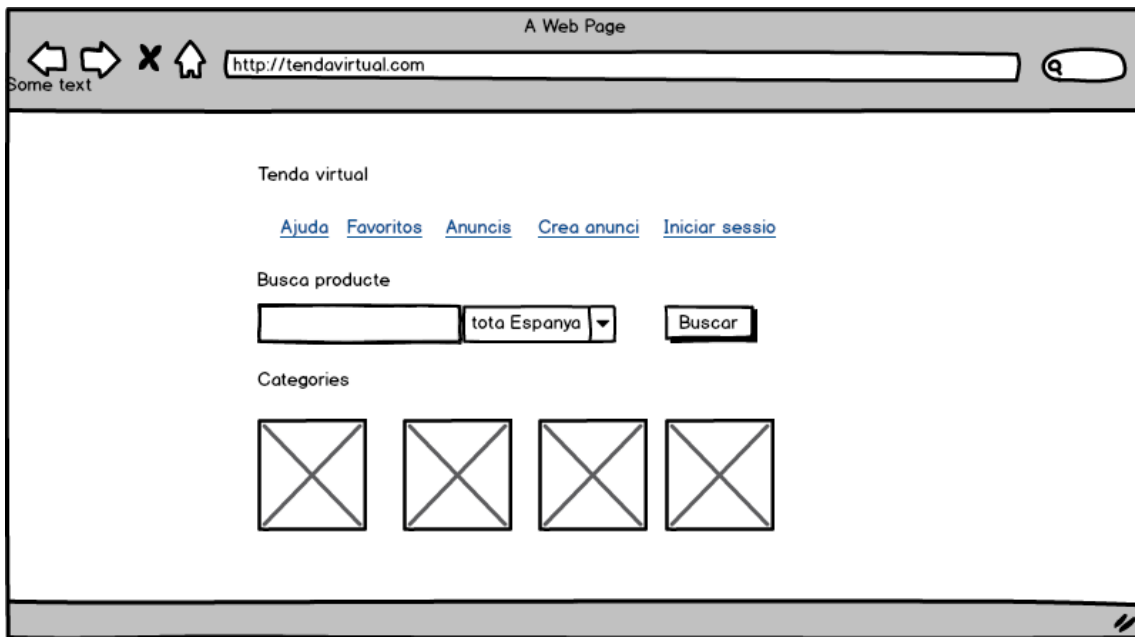
L'arquitectura que es seguirà en aquest projecte és la de Model-Vista-Controlador (MVC). El model es definirà al fitxer *models.py*, propi del *framework Django*, la vista estarà formada pels *templates* del programa i les accions de control s'implementaran, principalment, al fitxer *views.py*. Els mètodes d'aquest darrer fitxer obtindran les dades necessàries del model i les enviaran a la vista. Posteriorment, gestionaran la resposta d'aquesta fent les operacions pertinents i aplicant els canvis necessaris al model. A més, a *views.py* es declararan els mètodes que permetran interactuar dinàmicament amb la vista intercanviant dades durant la seva mostra, és a dir, els que rebran les crides d'AJAX iniciades als *templates*.

Per tant, el model i el controlador s'implementaran amb el llenguatge de programació *Python* i, en canvi, els elements de la vista es codificaran en *HTML*, *CSS* i *JavaScript*.

### 4.2 Prototipat de la web

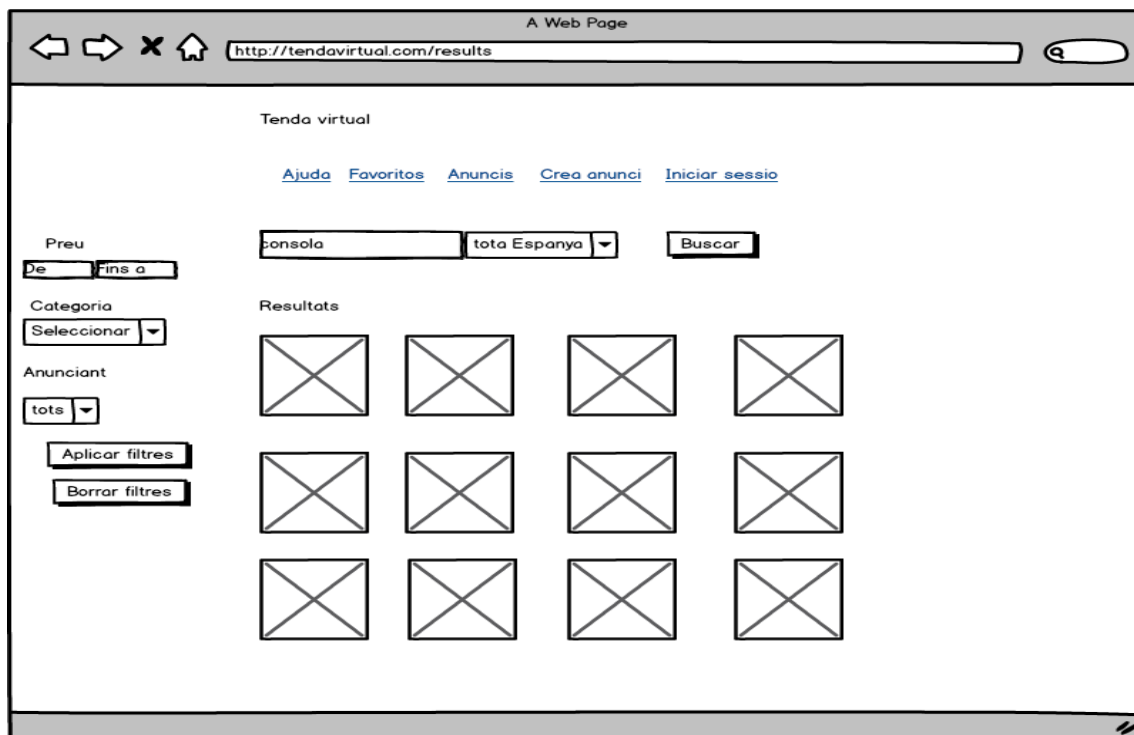
Amb l'eina *Balsamiq*s'han realitzat, durant la fase de disseny, els següents prototips d'algunes de les pantalles que tindrà el lloc web.

La primera part es dissenyar el contingut que se mostra al usuari, mitjançant un wireframe o maqueta. Un wireframe esquematitza el disseny de la pagina, però falta de estil ja que el principal objectiu de aquest es mostrar la organització dels elements de la web. A continuació es mostren unes figures amb la maqueta de algunes pagines de l'aplicació.



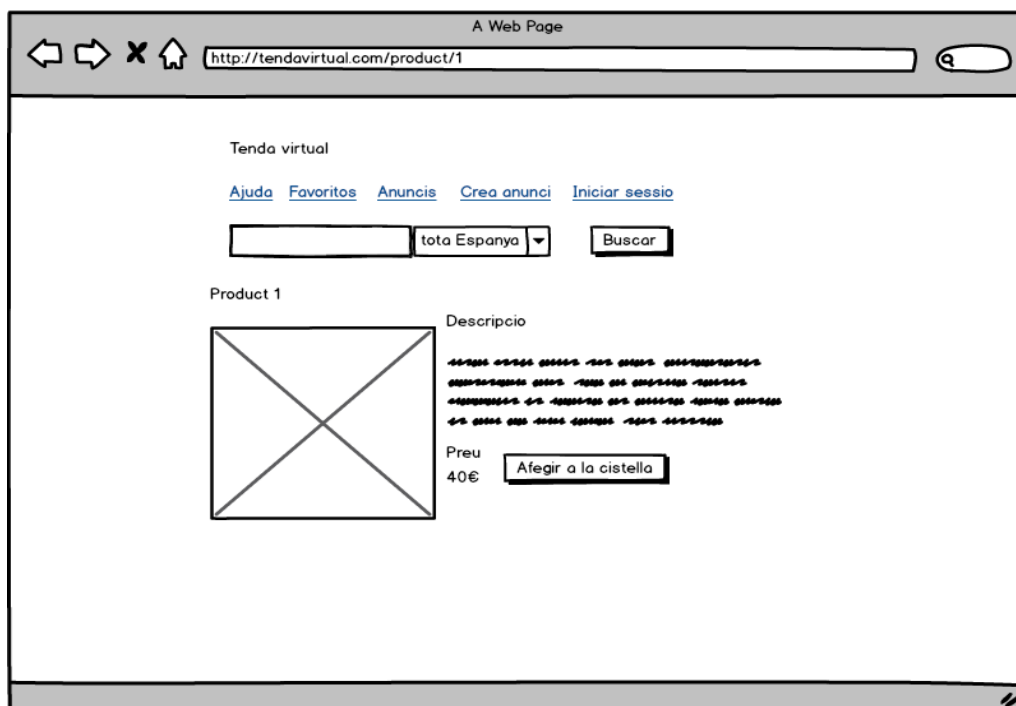
**Figura 4: Pagina principal de l'aplicatiu web.**

A la figura 4 podem identificar el disseny de la pagina principal, on passaran la majoria de peticions d'usuari amb enllaços en especial iniciar sessió, un buscador de productes on el usuari podrà fer les consultes personalitzades i per últim la selecció d'una categoria en concret que mostrarà tots els productes relacionats amb la categoria.



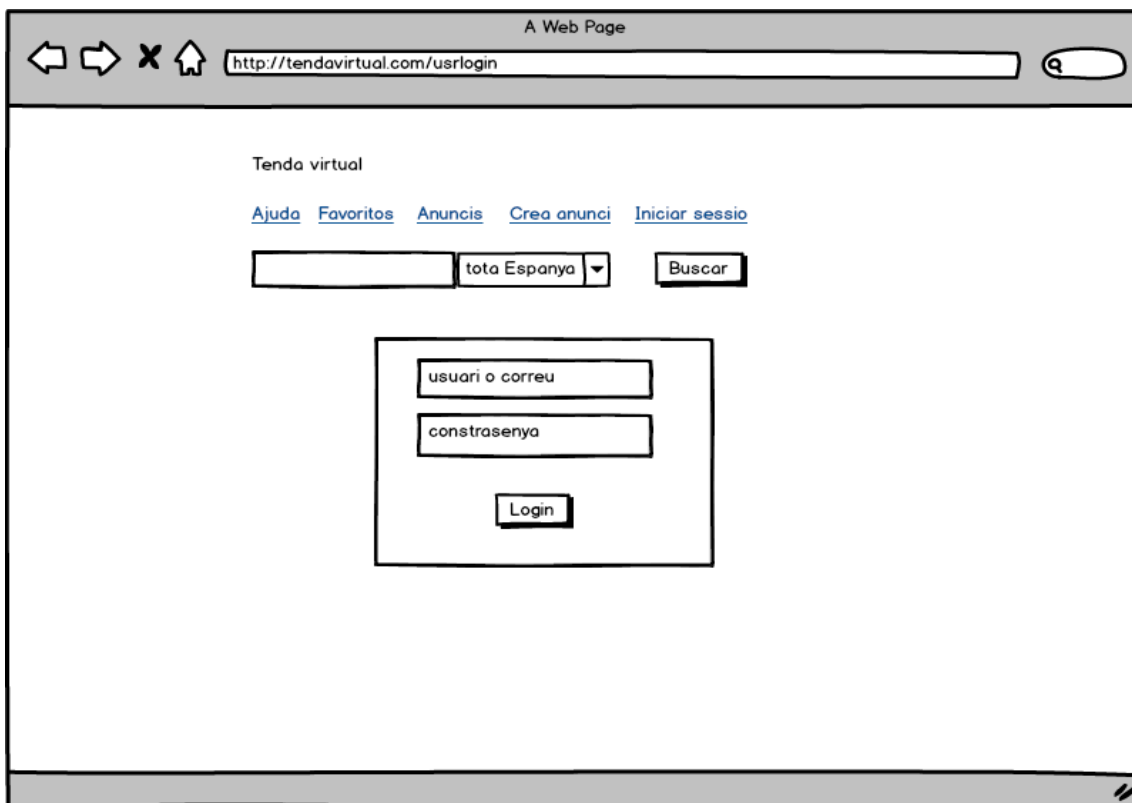
**Figura 5 : Pagina de resultats de una cerca.**

A la figura 5 podem contemplar el resultats de una cerca, que ha fet el usuari un cop ha pitjat el boto Buscar, la pagina de resultats compta, una informació de productes relacionats amb la paraula clau que ha ficat el usuari i unes opcions de consulta que permetran filtrar el continguts de resultats segons el preu, categoria i anunciant.



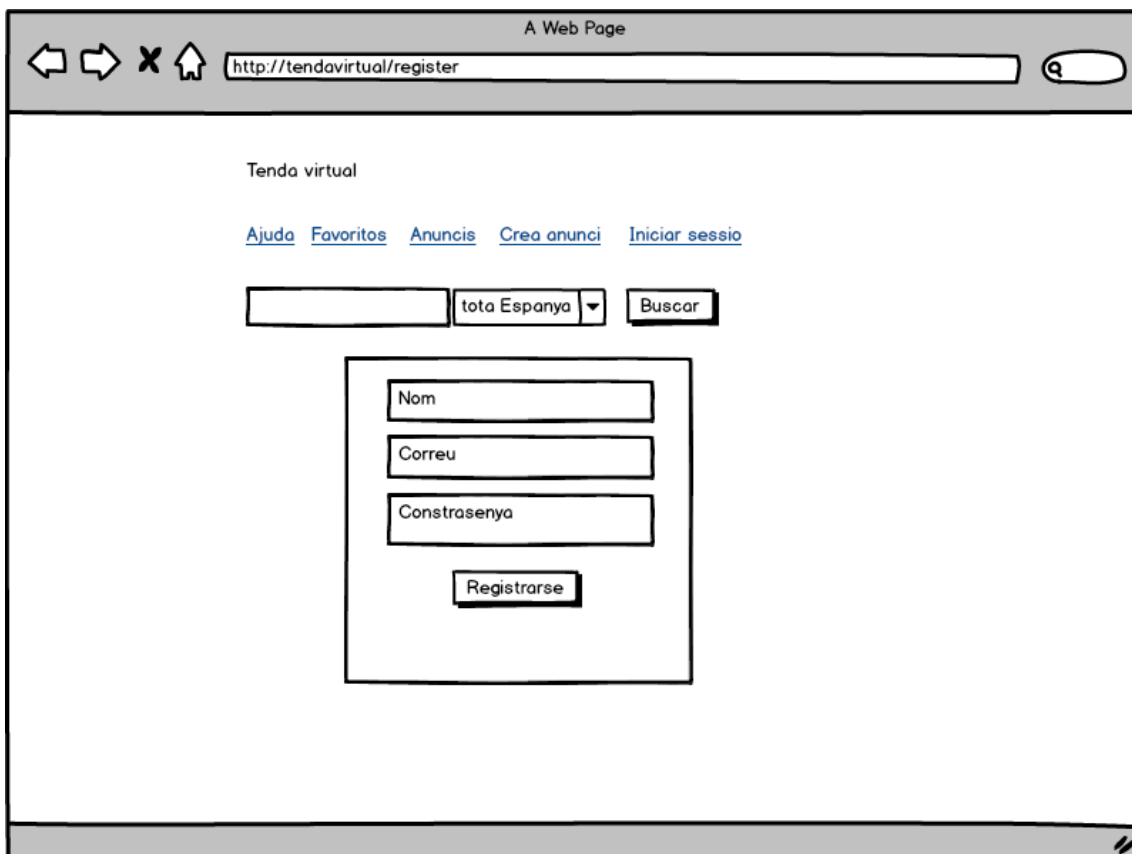
**Figura 6: Visualització de un producte**

A la figura 6 es pot contemplar el disseny de la visió de un producte, un cop el usuari a pitjat damunt de un producte, que inclou la descripció del producte, preu i la opció de Afegir a la cistella.



**Figura 7: Visualització de la pagina de login.**

A la figura 7 es pot veure el disseny de la pagina de login, on el usuari es podrà loguejar i accedir a altres pagines de continguts.



**Figura 8 Visualització de la pagina de registre.**

A la figura 8 com es pot contemplar, es el disseny de la pagina de registre de usuari, on l'usuari podrà registrar-se i crear-se un compte a la pagina.

Les figures que acabem de veure, son casos d'usos on podran accedir tan el usuari com el administrador.



A Web Page

http://tendavirtual.com/admin/createproduct

Productes Categories Pagaments Anuncis

Nom

Descripcio

Foto

Categoria -Ninguna-

Preu

Examinar

Crear producte

Figura 9: Visualització de creació de un producte .

A Web Page

http://tendavirtual.com/admin/categories

Productes Categories Pagaments Anuncis

Id	Nom categoria	Num productes
12	Inmobles	20
22	Electronica	200
43	Animals	145

Afegir categoria

Figura 10: Visualització de categories.

### 4.3 Diagrama de classes

Ara mostrarem les classes que s'utilitzaran per implementar el projecte, amb el seus atributs i funcions, i com es relacionen entre elles.

El **framework Django** permet definir aquestes classes mitjançant el fitxer **models.py** i amb les comandes **makemigrations** i **migrate** permet generar i actualitzar les bases de dades de forma automàtica. Per tant, no es necessari el posterior disseny i l'elaboració del diagrama de la base de dades.

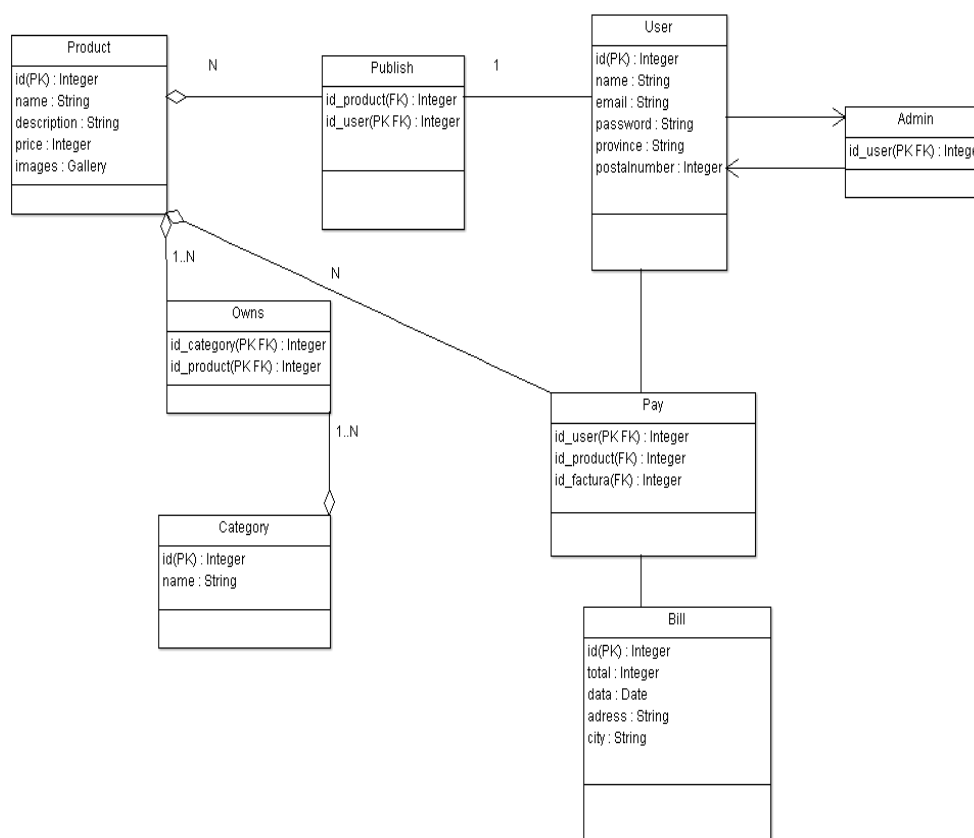


Figura 11: Diagrama de classe

## 4.4 Els templates

Els següents templates permetran carregar les diferents pantalles del programa. Tot ells s'hauran de titular de manera que l'usuari pugui identificar la pantalla en la que es troba. Amb la finalitat de no ocupar un espai que pot ser necessari per la mostra de múltiples continguts, sobretot en pantalles d'una dimensió menor, aquest títol es trobarà a la part superior de la finestra del navegador on es carregui el lloc web o en les mateixes pestanyes d'aquest.

Nom del template	Descripció de la pagina que carrega el template	Perfils
<b>Index</b>	Pàgina inicial de la web. Permetrà escollir varies opcions, principalment buscar productes i la d'iniciar sessió.	Usuari extern
<b>Buscar</b>	<p>Pàgina que permet introduir paràmetres a una barra de text per iniciar una consulta als productes disponibles que hi ha a la tenda.</p> <p>També, hi ha un menú que permet filtrar el productes per categories i preu.</p> <p>Els productes apareixeran en ordre de data de creació, així per poder saber quin es el producte mes nou a la tenda.</p>	Usuari extern
<b>Login</b>	Pàgina que permet al usuaris iniciar sessió. Demanarà el nom de usuari i la contrasenya.	Usuari extern
<b>Crear compte</b>	Pàgina que permet registrar al usuaris externs, demanant les dades necessàries per fer el registre.	Usuari extern
<b>Visualitzar producte</b>	<p>Pàgina que permet visualitzar les dades del productes: nom del producte, descripció ,preu i les imatges del producte.</p> <p>L'usuari dispondrà de l'opció , afegir al carro que s'explicarà mes endavant el seu us.</p>	Usuari extern
<b>Crear anunci</b>	Pàgina que permet crear un anunci d'un producte. L'usuari introduirà títol, descripció, preu, imatges i categoria.	Usuari registrat

<b>Editar anunci</b>	Pàgina que permet editar un anunci creat. Pot modificar títol, descripció, preu, imatges i categoria.	Usuari registrat
<b>Eliminar anunci</b>	Pàgina que permet eliminar un anunci, ja creat. Apareix una opció per fer aquesta acció.	Usuari registrat
<b>Carro</b>	Pàgina que mostrarà els productes seleccionats anteriorment a la pàgina de visualitzar producte. El carro mostra tota la informació de la compra, que l'usuari a anat introduint.	Usuari registrat
<b>Afegir al carro</b>	Pàgina que permet al usuari afegir el producte al carro de compra.	Usuari registrat
<b>Eliminar del carro</b>	Pàgina que permet al usuari eliminar el producte del carro de compra.	Usuari registrat
<b>Editar perfil</b>	Pàgina que permet al usuari la informació del perfil d'usuari. Podrà editar el username, correu, contrasenya, província i el codi postal.	Usuari registrat
<b>Admin</b>	Pàgina on mostrarà la informació del administrador del lloc web amb el seu correu.  El administrador podrà gestionar els clients, productes i pagaments.  On es mostraran les dades en una taula.	Administrador
<b>Crear clients</b>	Pàgina que permet crear nous usuaris com a administrador.	Administrador
<b>Editar clients</b>	Pàgina que permet editar la informació del clients com a administrador.	Administrador
<b>Eliminar clients</b>	Pàgina que permet eliminar els clients de la base de dades, com a administrador.	Administrador
<b>Crear productes</b>	Pàgina que permet crear productes com a administrador.	Administrador
<b>Editar</b>	Pàgina que permet editar la informació dels	Administrador

<b>productes</b>	productes com a administrador.	
<b>Eliminar productes</b>	Pàgina que permet eliminar productes de la base de dades com a administrador.	Administrador
<b>Crear pagament</b>	Pàgina que permet crear pagaments com a administrador.	Administrador
<b>Editar pagament</b>	Pàgina que permet editar la informació de pagament com a administrador.	Administrador
<b>Eliminar pagament</b>	Pàgina que permet eliminar pagaments com a administrador.	Administrador

# 5.Implementació

## 5.1 Esquema organitzatiu de fitxers

Els directoris on es troben els diferents fitxers del projecte segueixen la següent estructura, establerta pel *framework Django*:

- **Rummitenda:** Conté els fitxers *python* on s'estableixen els criteris de configuració general que s'apliquen a totes les aplicacions desenvolupades. En aquest cas, només es desenvolupa l'aplicació de reserves.

Per exemple, en els fitxers d'aquest directori es determina la configuració de l'idioma del lloc web, la del gestor de bases de dades utilitzat o la del servidor de correu des del qual s'enviaran els *mails*.

- **Rummi:** Conté diversos fitxers *python* tals com el que desenvolupa les funcionalitats de control o en el que es defineixen les diferents *urls* del lloc web. A més, en aquest directori es troben els següents subdirectoris:

- **Templates:** Conté els fitxers *html* que carreguen l'estructura de les diferents pàgines del lloc web. A més, a la carpeta *textos* es troben els fitxers que contenen el cos dels diferents correus que el programa pot enviar.
- **Static:** Conté les imatges que carreguen les pàgines del lloc web, les diferents plantilles que permeten generar els fulls de reserva o els llistats que el programa permet extreure, i els documents pujats pels usuaris al sistema. A més, el software s'ha configurat per tal que aquest directori sigui accessible des dels altres *templates* mitjançant la sentència *static* o des dels fitxers amb extensió *.py* important els *settings* (la configuració) i cridant la constant *STATIC\_ROOT*.
- **Migrations:** Conté els fitxers que genera *Django* quan es realitzen les migracions a la base de dades a través de les comandes *makemigrations* i *migrate*.

## **5.2 El model de dades**

En aquest projecte django usem MySQL, es un sistema de gestió de bases de dades relacional, es un conjunt de programes que permeten el emmagatzematge, modificació i extracció de la informació en una base de dades. Els usuaris podran accedir a la informació emprant eines específiques de consulta i generació de informes.

### **5.2.1 El model relacional**

El model relacional per al modelat i la gestió de base de dades, es un model de dades basat en la lògica de predicats i en la teoria de conjunts.

La seva idea fonamental es el us de relacions. Aquestes relacions podrien considerar en forma lògica com a conjunt de dades nombrats tuples.

En aquest projecte s'ha emprat el model relacional, que es el model mes emprat en la actualitat per modelar problemes reals i administrar dades dinàmicament.

### **5.2.2 Les relacions**

#### **5.2.2.1 ManyToManyField**

Permet establir la relació molts a molts entre dues classes del model.

#### **5.2.2.2 OnetoOneField**

Permet establir la relació un a un entre dues classes del model. Per defecte, s'assigna el paràmetre `on_delete=models.CASCADE` i aleshores, quan un objecte s'esborra de la base de dades també s'esborra, automàticament, el que te relacionat.

### 5.2.2.3 ForeignKey

Permet establir la relació un a molts entre dues classes del model. Per defecte s'assigna el paràmetre `on_delete=models.CASCADE` i, aleshores, quan l'objecte unitari de la relació s'esborra de la base de dades, automàticament també s'esborren els que té relacionat.

### 5.2.3 Classes

Un altre aspecte que cal mencionar és el dels tipus d'atributs utilitzats en el projecte:

*BooleanField*, *CharField*, *DateField*, *DateTimeField*, *EmailField*, *FloatField*, *PositiveIntegerField* i *TimeField*. El tipus *DateTimeField* s'ha assignat als camps pels quals interessa guardar tant la data, com l'hora, els minuts i els segons, a diferència de *DateField* que només guarda la primera. Principalment, els atributs *DateTimeField* són els utilitzats pels atributs implementats per contenir la data de creació dels objectes perquè, en casos com el dels productes, l'usuari pot crear-ne més d'una de manera pràcticament immediata. Aleshores, la característica de poder arribar a comparar-les per les hores, minuts i segons de l'instant en el qual s'han creat permet al sistema mostrar-les correctament ordenades.

Als esmentats camps de la data de creació se'ls hi assigna el valor corresponent de manera automàtica gràcies al paràmetre `auto_now_add=True` amb el que es declaren. Al projecte també s'utilitzen altres paràmetres en la declaració dels *fields* que permeten, per exemple, indicar la màxima longitud d'una cadena de caràcters o establir si el camp es pot deixar buit o no.



## 5.2.4 Les consultes o queries

A continuació, es mostra un llistat dels mètodes utilitzats en aquest projecte per fer les consultes a la base de dades i les modificacions dels seus valors. Aquests mètodes són propis del model del *framework Django* i permeten realitzar les consultes d'SQL mitjançant una simple crida, la qual té sempre la mateixa estructura:

*nom de la classe del model o de la taula de la base de dades a la que es vol accedir + “.objects” + mètode o mètodes que es criden precedits cada un per un punt*

Els mètodes esmentats són els següents:

- **All:** Retorna tots els objectes de la taula de la base de dades. No se li passa cap paràmetre.
- **Filter:** Retorna tots les objectes que compleixen les condicions passades per paràmetre.
- **Get:** Retorna un únic objecte. Si troba més d'un objecte pels paràmetres passats, es produeix una excepció.
- **Exclude:** Exclou del resultat de la consulta tots aquells objectes que compleixen les condicions que se li passen per paràmetre.
- **Exists:** Retorna *True* si la consulta realitzada ha trobat almenys un objecte. Retorna *False* si no n'ha trobat cap.
- **Order by:** Permet ordenar el resultat de la consulta segons els paràmetres passats.
- **Values list:** Retorna una llista de *tuples* on cadascuna d'aquestes conté els atributs dels objectes retornats per la consulta que es demanen per paràmetre. Si es crida el mètode passant-li *flat=True*, retorna directament una llista que comprèn tots aquests atributs sense que estiguin compresos en *tuples*.
- **Distinct:** Elimina els objectes duplicats que hi pugui haver en el resultat de la consulta.  
Només es pot aplicar si aquest resultat es troba en forma de llista. Per aquest motiu, en el projecte només s'aplica després d'haver utilitzat el mètode *values\_list* amb el paràmetre *flat=True*.

- **Save:** Permet guardar els canvis realitzats en els atributs d'un objecte. Aquest mètode s'utilitza en el projecte per crear una nova instància d'una classe i per aplicar les modificacions o edicions pertinents a un objecte creat amb anterioritat.
- **Delete:** Permet esborrar els objectes de la base de dades.
- **Add:** Permet afegir nous objectes a les relacions *many to many*.
- **Clear:** Permet esborrar totes les relacions *many to many* però sense eliminar els objectes relacionats. Aquest mètode s'utilitza en el projecte per actualitzar aquestes relacions de molts a molts, ja que primer es buiden i, seguidament, es relacionen tots els objectes que ho han d'estar. D'aquesta manera, s'evita haver d'analitzar quins objectes s'han d'eliminar de la relació entre classes i quins s'han d'afegir que no ho estiguin ja.

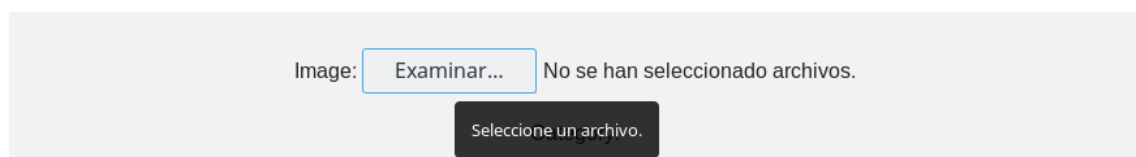
## 5.3 La gestió de documents

El lloc web només guarda al servidor els arxius relacionats amb un producte que els usuaris del programa pugen. Els que pot generar el sistema amb la informació de què disposa són generats cada cop que es demanen.

### 5.3.1 Generació, pujada i visualitzar imatges.

Al directori *static* es troben les imatges del producte que el usuari a penjat a l'hora de crear anuncis.

La pujada de documents es porta a terme mitjançant el camp *input* del tipus *file*. Aquest camp permet la pujada de fitxers limitant els formats acceptats. El programa accepta els següents formats: *.jpg*, *.png*, *.bmp*, *.gif*.



**Figura 12: Pujada de fotos del producte**

## **5.4 Interfície**

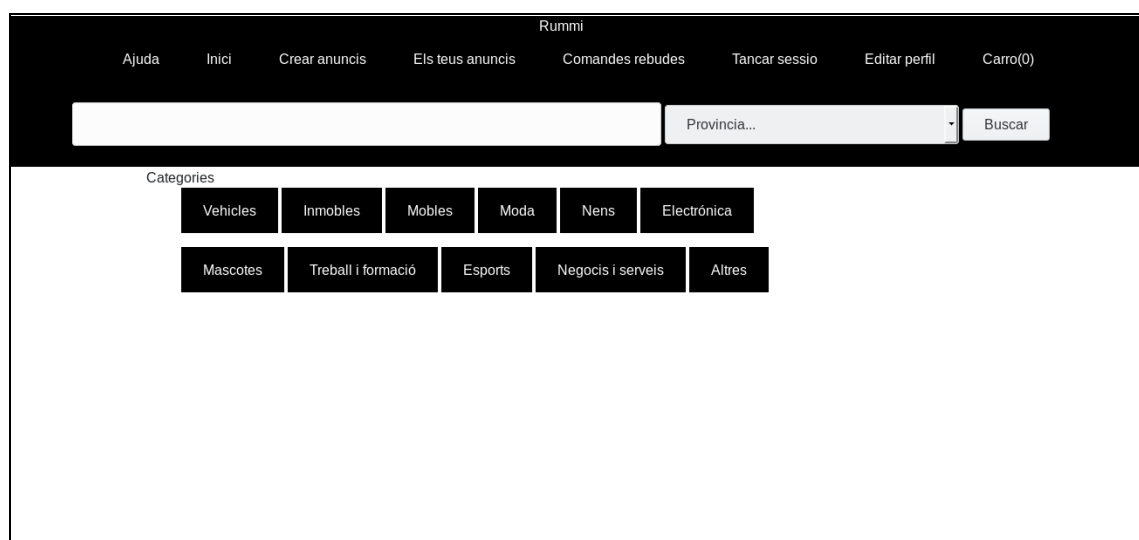
### **5.4.1 La aplicació client**

Per implementar aquesta interfície, s'ha utilitzat HTML i Python, per a tota la aplicació client, amb això hem pogut desenvolupar totes les accions del digrama de cas d'usos. Moltes de les accions es mencionaran aquí de nou, per explicar el seu breu funcionament.

Per començar el directori “tenda” compte una carpeta que guarda els HTML nombrada html. Cada HTML que compte aquesta carpeta fa referencia a unes de les accions, que construeix la aplicació.

Les views rebran les peticions quan es crida la url, una url única per a cada acció, tenint en compte això hem dissenyat una web comercial. Un cop ates la petició, es comença a buscar la view, que generarà una resposta en cada acció.

Les html s'encarregaran de dissenyar i rebre les respostes tal com li indiqui la view.



### **5.4.2 Les views**

Una view és on es fica la “lògica” de la nostra aplicació. Una funció de vista o vista curta és simplement una funció de Python que pren una sol·licitud web i retorna una resposta web. Aquesta resposta pot ser el contingut HTML d'una pàgina web, o una redirecció, un error 404 o un document XML o una imatge. . . o qualsevol cosa, en realitat. La vista en si conté qualsevol lògica arbitrària necessària per retornar aquesta resposta. Aquest codi pot viure en qualsevol lloc que vulgueu, sempre que es trobi en la vostra ruta de Python. No hi ha cap altre requisit. Per tal de posar el codi en algun lloc, la convenció és posar vistes en un fitxer anomenat `views.py`, ubicat al directori del nostre projecte o aplicació.

Per exemple tenim la view `login` que permet accedir a la web passant-li unes dades introduïdes mitjançant un form i fent consulta a la nostra base de dades, per verificar si el usuari existeix.

Aquesta rep un objecte `request`, que disposa de uns mètodes per ajudar-nos a desenvolupar el codi.

A mode d'exemple, a continuació es mostra breument la implementació d'un `request.GET` que conté un diccionari que proveix paràmetres GET.

```

#Permet crear productes com a usuari
@login_required(login_url='login')
def create_product(request):
    #dataset = Product.objects.filter(id=id)[0]
    if request.method=='POST':
        print "POST-Create_product"
        form = ProductForm(request.POST, request.FILES)
        if form.is_valid():
            n= form.cleaned_data.get('name')
            img = form.cleaned_data.get('image')
            desc = form.cleaned_data.get('description')
            p = form.cleaned_data.get('price')
            ca = form.cleaned_data.get('category')
            cat = Category.objects.get(name=ca)

            new_data = Product.objects.create(name=n, description=desc, price = p)
            for count,x in enumerate(request.FILES.getlist('image')):
                handle_uploaded_file(x, str(x))
                Images.objects.create(post=new_data, image=str(x))
            u = UserProfile.objects.get(username=request.user)
            Publish.objects.create(user=u, product=new_data)
            Owns.objects.create(category=cat, product=new_data)
            new_data.save()

            return render(request, 'category.html')
        else:
            form = ProductForm()
    return render(request, 'create_product.html',{'form': form,})

```

Figura 14: Una view que s'encarrega de crear un producte.

### 5.4.3 Forms

Els forms son camps del qual permetren entrar dades de tot tipus.

Hem descrit breument els formats HTML, però un HTML <form> és només una part de la maquinària requerida.

En el context d'una aplicació web, "formulari" pot referir-se a aquest formulari <HTML>, o al format de Django que el produeix, o les dades estructurades retornades quan s'envia o a la col·lecció de treball de punta a punta d'aquestes parts.

El nucli d'aquest sistema de components és Django's Form class. De la mateixa manera que un model de Django descriu l'estructura lògica d'un objecte, el seu comportament i la manera com se'ns representen les parts, una classe de formulari descriu una forma i determina com funciona i apareix.

De manera semblant, que els camps de la classe del model es classifiquen als camps de la base de dades, els camps de la classe de formulari s'apliquen als elements <input> d'HTML. (Un ModelForm assigna els camps de la classe del model als elements HTML <input> a través d'un formulari; això és el que l'administrador de Django està basat en).

Els camps d'una forma són classes de si mateixes; gestionen les dades del formulari i realitzen la validació quan es presenta un formulari. Un `DateField` i un `FileField` manegen tipus de dades molt diferents i han de fer coses diferents amb ell.

Es representa un camp de formulari a un usuari del navegador com a «widget» HTML: una maquinària d'interfície d'usuari. Cada tipus de camp té una classe de widget predeterminada adequada, però aquests poden ser anul·lats segons sigui necessari.

```
class CartAddProductForm(forms.Form):
    update = forms.BooleanField(required=False, initial=False, widget=forms.HiddenInput)

class Consultaform(forms.Form):
    consulta = forms.CharField(label='consulta', required=False, max_length=100)
    provincia = forms.ChoiceField(label='provincia', choices=PROVINCES, required=False)

class Filtresform(forms.Form):
    preu = forms.IntegerField(required=False)
    finsa = forms.IntegerField(required=False)

class SignUpForm(forms.Form):
    name = forms.CharField(max_length=100)
    password = forms.CharField(max_length=100)
```

**Figura 15: Uns formularis que s'encarreguen de recollir dades.**

#### 5.4.4 Les urls

Per dissenyar URL per a una aplicació, creeu un mòdul Python informalment anomenat `URLconf` (configuració de l'URL). Aquest mòdul és codi pur de Python i és un mapatge entre les expressions de la ruta d'accés URL a les funcions de Python (les nostres visualitzacions).

Aquest mapa pot ser tan curt com sempre que sigui necessari. Pot fer referència a altres assignacions. I, perquè és codi pur de Python, es pot construir dinàmicament.

Django també proporciona una manera de traduir URL segons el llenguatge actiu.

```

from django.conf.urls import url, include
from django.contrib import admin
from tenda.views import RegisterUIView, LoginView, LogoutView
from tenda import views
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    #url(r'^admin/', admin.site.urls),
    url(r'^$', views.buscar, name='buscar'),
    url(r'^filtra', views.filtro, name='filtro'),
    url(r'^search_category', views.search_category, name='search_category'),
    url(r'^create_product', views.create_product, name='create_product'),
    url(r'^product_list/$', views.product_list, name='product_list2'),
    url(r'^product_list/(?P<product_id>\d+)/edit/$', views.edit_product, name='edit_product2'),
    url(r'^product_list/(?P<product_id>\d+)/eliminate/$', views.delete_product, name='eliminate_product2'),
    |
    #Payment
    url(r'^payment$', views.payment, name='payment'),
    url(r'^payment_last$', views.payment, name='payment_last'),

    url(r'^payment/success$', views.success_payment, name='success_payment'),
    url(r'^payment/canceled$', views.unsuccess_payment, name='unsuccess_payment'),

    url(r'^paypal/', include('paypal.standard.ipn.urls')),
    url(r'^tickets$', views.tickets_list, name='tickets_list'),

    url(r'^product/(?P<id_product>\d+)', views.product, name='product'),
    url(r'^profile', views.profile_edit, name='profile'),

```

**Figura 16: Urls amb la seva view corresponent.**

### 5.4.5 Carro

Per dissenyar el carro hem creat una classe cart.py, que permet el us d'un carro a partir d'una variable Django request.session on allí es guardaran el productes, simplement es una llista que te implementats uns mètodes per poderla gestionar.

A mes funciona de forma independent de la aplicació web, osigui no molesta a les peticions que fa.

**Figura 17: Vista preliminar del carro de compra.**

### 5.4.6 Paypal

Per implementar el sistema de pago hem tingut que recórrer a una API especial de paypal, que compte els mètodes necessaris per poder programar un sistema de pagament. Python-paypal es la Api que hem emprat, esta escrit en Python i ens permet comunicar-nos amb la companyia paypal. Per comunicar-se amb la pagina s'utilitza un form proporcionat per la api, simplement es un boto que enllaça a la pagina de paypal amb la informació proporcionada pel form, que permet fer la transacció desitjada.

Per fer un pagament el usuari haurà de elegir el productes que hi ha disponibles, després el usuari pitjarà amb el boto 'Add to Cart', per guardar el producte al carro, així tants productes com vulgui l'usuari. El usuari quan hagi completat de elegir productes, anirà a la opció carro que esta al menú de dalt a la dreta on hi podrà accedir, així doncs arribem a la part que el usuari comença el pagament. Primer veurà un boto 'Tramitar pagament' que polsara amb el ratolí, després d'això començarà el procés de pagament amb un formulari que li demana les dades de enviament, a on vol que ho envio, després de replanar el formulari pitja el boto 'Pagar' llavors el usuari arriba al últim pas que es el mètode de pagament, on podrà indicar quin mètode emprar per acabar la comprar, en aquest cas nomes s'utilitza paypal.

Després de pitjar el boto paypal, el usuari anirà a la pagina amb la informació de la compra, tindrà que crear-se un compte. Un cop creat podrà procedir amb el pagament de la pagina, li indicarà el preu a pagar i el usuari sol tindrà que pitjar 'Pagar'.

Durant les proves s'ha emprat un compte de exemple per a dur a terme les transaccions fictícies que permet fer la API de paypal.



Rummi

[Ajuda](#) [Inici](#) [Crear anuncis](#) [Els teus anuncis](#) [Comandes rebudes](#) [Tancar sessio](#) [Editar perfil](#) [Carro\(0\)](#)

### Pas 2: Dades de enviament

Adreça:

Ciutat/Població:

Provincia:

Codi Postal:

**Figura 18: Vista preliminar del procés de pagament en aquest cas les dades de enviament.**

Rummi

[Ajuda](#) [Inici](#) [Crear anuncis](#) [Els teus anuncis](#) [Comandes rebudes](#) [Tancar sessio](#) [Editar perfil](#) [Carro\(0\)](#)

### Pas3: Metode de pagament


Destinatari: aurelia

Adreça: Alcalde Rovira Roure 105

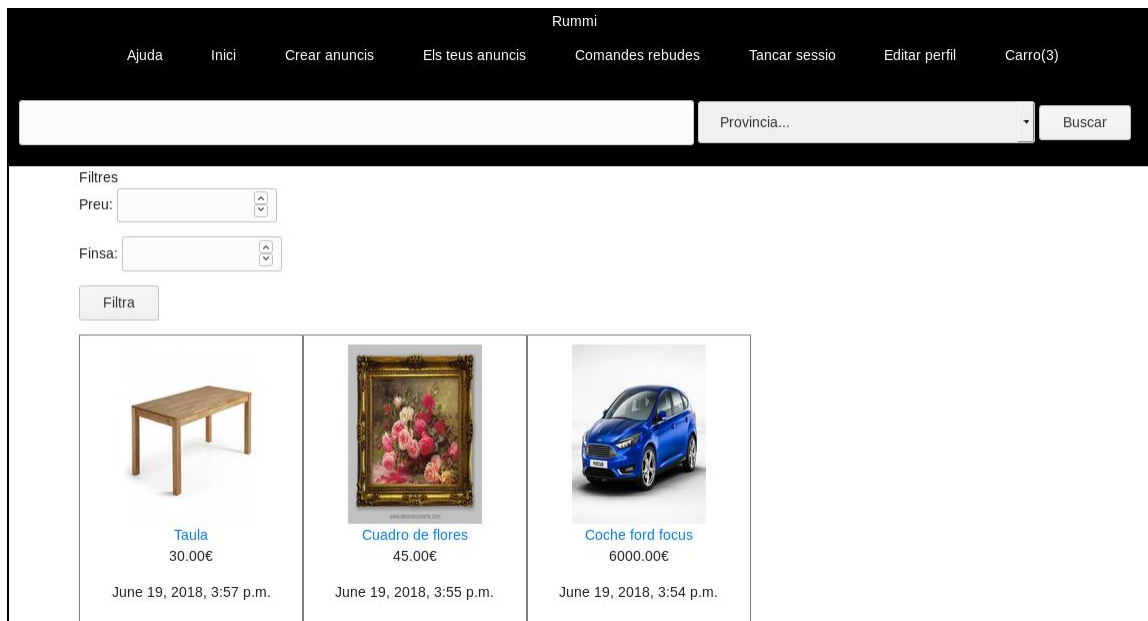
Ciutat/població: Lleida

Provincia: lleida

Codi postal: 25198



**Figura 19: Selecció del mètode de pagament.**



**Figura 20: Selecció d'un producte a partir del boto "Buscar"**

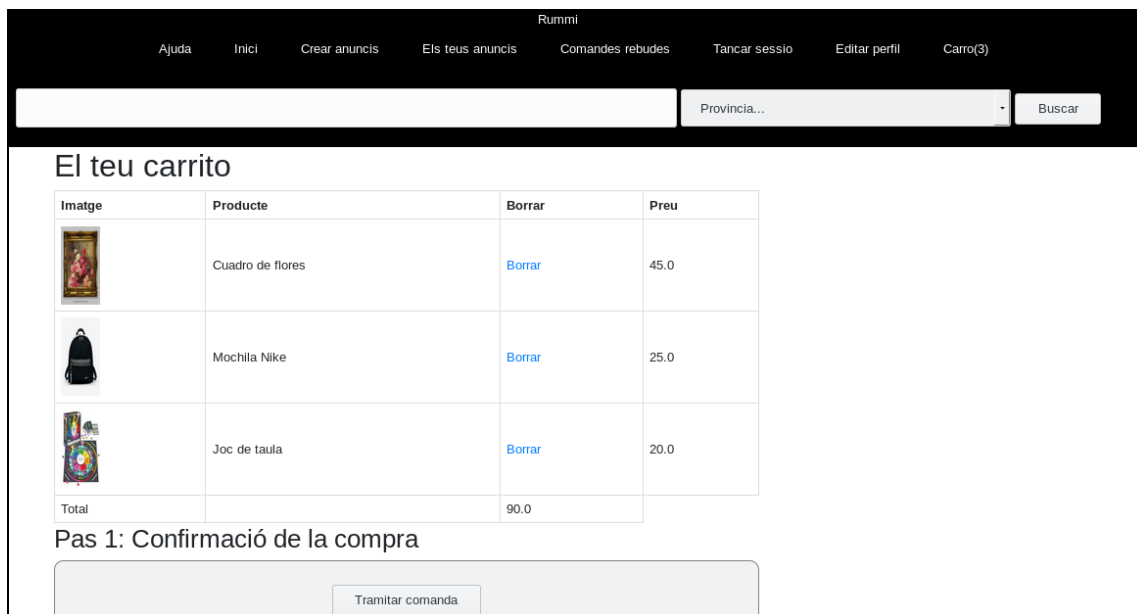
### 5.4.7 Selecció d'un producte

En el cas de la implementació de la interfície, hem desenvolupat un mètode de selecció de producte per al usuari, primerament hem dissenyat la pagina en format HTML, per donar-li cert aspecte a la pagina.

La figura 20 mostra una sèrie de productes un cop pitjat el boto "Buscar", que el usuari podrà elegir pitjant el títol del producte, un cop pitjat mostrarà la informació del producte a la pantalla, on disposarà d'una altra opció que es "Add to cart", que polsarà l'usuari, s'afegirà un producte al carro de la compra que disposa la aplicació client.

Un cop que l'usuari haguí decidit acabar amb la compra, polsarà al menú el boto "Carro(0)", on es redirigiran, a una llista de productes on indicarà el preu, títol del producte, imatge i la opció "Borrar", mes avall el procés de pagament de la pagina i el boto "Tramitar comanda".




A la figura 21 es mostra el contingut del carro de compra que hem fet amb un usuari, amb tots els seu elements.



**Figura 21: Carro de compra amb productes seleccionats.**

### **5.4.8 Els teus productes.**

Al menú de dalt, si pitgem “Els teus anuncis” surt una pagina de productes que te el usuari. Es una taula on mostra imatge, nom del producte, dos opcions de usuari el preu. Te com a objectiu administrar els productes que el usuari ha publicat. Hi ha dues opcions: “Editar” permet modificar la informació del producte i “Borrar” permet eliminar la informació del producte.

Rummi				
<a href="#">Ajuda</a>	<a href="#">Inici</a>	<a href="#">Crear anuncis</a>	<a href="#">Els teus anuncis</a>	<a href="#">Comandes rebudes</a>
			<a href="#">Tancar sessio</a>	<a href="#">Editar perfil</a>
			<a href="#">Carro(0)</a>	
<input type="text"/>			Provincia...	<input type="button" value="Buscar"/>
El teus anuncis				
Imatge	Producte	Editar	Borrar	Preu
	Nintendo 64	<a href="#">Editar</a>	<a href="#">Borrar</a>	50.00
	Portatil hp	<a href="#">Editar</a>	<a href="#">Borrar</a>	400.00
	Cascos de musica	<a href="#">Editar</a>	<a href="#">Borrar</a>	35.00

**Figura 22: El anuncis que ha penjat el usuari.**

Rummi				
<a href="#">Ajuda</a>	<a href="#">Inici</a>	<a href="#">Crear anuncis</a>	<a href="#">Els teus anuncis</a>	<a href="#">Comandes rebudes</a>
			<a href="#">Tancar sessio</a>	<a href="#">Editar perfil</a>
			<a href="#">Carro(3)</a>	
<input type="text"/>			Provincia...	<input type="button" value="Buscar"/>
Comandes rebudes				
Data	Adreça	Ciutat/Població	Preu	
June 5, 2018, 4:44 p.m.	Alcalde Rovira Roure 105	Lleida	4000.00	
June 5, 2018, 4:46 p.m.	Alcalde Rovira Roure 105	Lleida	500.00	

**Figura 23: Les comandes o pagaments rebuts a la web.**

#### 5.4.9 Les comandes.

Son notificacions de sistema, que confirmen que s'ha a dut a procés el pagament de manera satisfactòria. Guarda al usuari, una comanda que representa la compra amb les seves dades corresponents. Esta ordenat per data de comanda. Per així dur a terme una administració dels pagaments.

A la figura 23 es pot contemplar dues comandes completades, amb la data, adreça, ciutat i preu.

## 6. Conclusions

Considero que els objectius específics d'aquest projecte de fi de grau s'han assolit gràcies als diferents coneixements que he adquirit, tant teòrics com pràctics, relacionats amb la implementació d'un lloc web mitjançant el *framework Django* i altres eines tals com el gestor *MySQL* de bases de dades *SQL*. A més, he pogut comprovar la importància de realitzar una correcta anàlisi dels requeriments del programa i de portar a terme un disseny acurat per tirar endavant satisfactòriament un projecte de desenvolupament de software, i més encara quan aquest té una mida considerable.

La tenda es unes de les matèries que m'ha interessat més, tot el procés que comporta a l'hora de crear-la, així com tindre certa experiència en programar en certes plataformes web i començar al món del màrqueting, els objectius de la tenda s'han completat satisfactòriament, el poder fer compres online mitjançant una web, el usuari té l'opció tan de comprador com de venedor.

El procés de compra s'ha agilitzat tan com ha sigut possible, degut a la seguretat de la web, hem tingut que afegir varis tràmits per completar la compra, com les dades de enviament i el mètode de pagament.

També he après a desenvolupar una aplicació web en solitari.

### 6.1 Futures ampliacions

Aquest projecte podria formar part d'un projecte de màrqueting més gran, com la millora del sistema de administrament de la web.

Incorporar nous sistemes de pagaments, com pagar amb Visa i altres plataformes online de pagament.

Enviament de missatges de completar la compra al correu del comprador.

Afegir més detalls com filtres personalitzats, que pugui filtrar productes amb més precisió.

Millorar el estil de pagina css, amb més detalls i ficar alguna cosa de AJAX ja que no ho hem pogut realitzar.

Implementar la tenda en una aplicació android. Per així promocionar la tenda en el mon dels mòbils.

## **Glossari**

**DJANGO:** És un *framework* de desenvolupament web escrit en *Python* que respecta el patró de disseny Model – Vista – Controlador.

**Framework:** És un entorn de treball que permet desenvolupar un programari de manera més senzilla, ja que aporta una sèrie de patrons i llibreries que formen l'esquema del software.

**JavaScript:** És un llenguatge orientat a objectes que permet implementar els *scripts* que atorguen un cert dinamisme als llocs webs.

**HTML:** *HyperText Markup Language* és un llenguatge de programació marcat que permet atorgar una descripció i unes característiques als textos que conté. Aquest llenguatge s'utilitza per a l'elaboració de pàgines webs.

**Templates:** És un conjunt d'arxius que permet determinar l'estructura i l'aspecte de les diferents pantalles d'un lloc web.

**Unicode:** És un estàndard internacional de codificació de caràcters en suports



## ***Bibliografia***

- Django Software Foundation. (2017). *Django documentation*. [en línia]. Recuperat el 27 de febrer de 2018 <https://docs.djangoproject.com/en/1.11/>
- Oracle Corporation. (2013). *MySQL 5.7 Reference Manual*. [en línia]. Recuperat el 5 de març de 2018 <https://dev.mysql.com/doc/refman/5.7/en/>
- Python Software Foundation. (2016). *Python 2.7.4 documentation*. [en línia]. Recuperat el 6 de març de 2018 <https://docs.python.org/2/>
- Django framework wikipedia. Recuperat el 24 de maig de 2018 [https://es.wikipedia.org/wiki/Django\\_\(framework\)](https://es.wikipedia.org/wiki/Django_(framework))
- Diagrama de gantt. Recuperat el 4 juny de 2018 <https://www.ganttproject.biz/>